INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Influence of an Interface Condition and Exact Linearization in the Newton Iterations for Two-Phase Flow in Heterogeneous Porous Media

Jennifer Niessner — Jean Roberts — Rainer Helmig

N° 4903

Aout 2003

THÈME 4

ISSN 0249-6399 ISRN INRIA/RR--4903--FR+ENG

Influence of an Interface Condition and Exact Linearization in the Newton Iterations for Two-Phase Flow in Heterogeneous Porous Media

Jennifer Niessner* , Jean Roberts† , Rainer Helmig‡

Thème 4 — Simulation et optimisation
de systèmes complexes
Projets Estime

Rapport de recherche n° 4903 — Aout 2003 — 93 pages

Abstract: Block heterogeneities have an important influence on macroscale multi-phase fluid flow and transport in porous media. We focus especially on the interface between different materials and linearization schemes in order to improve the efficiency of the numerical simulator.

Key-words: multi-phase flow, heterogeneous systems, numerical solution, interface condition, nonlinear solver

Rapport de stage de J. Niessner

* Projet Estime. Jennifer.Niessner@inria.fr.
† Projet Estime. Jean.Roberts@inria.fr.
‡ Institut für Wasserbau, Universität Stuttgart. Rainer.Helmig@iws.uni-stuttgart.de.

Unité de recherche INRIA Rocquencourt
Domaine de Voluceau, Rocquencourt, BP 105, 78153 Le Chesnay Cedex (France)
Téléphone : +33 1 39 63 55 11 —Télécopie : +33 1 39 63 53 30

Influence d'une condition d'interface et linéarisation exacte dans les itérations de Newton pour des écoulements diphasiques en milieu poreux hétérogènes

Résumé : Les hétérogénéités de bloc ont une influence importante sur l´écoulement et le transport multiphasique à l'échelle macroscopique en milieu poreux. On s'intéresse particulièrement à la mise en œuvre numérique d'une condition d'interface entre deux matériaux différents, ainsi qu'à l'amélioration de l'efficacité du solveur non-linéaire à l'aide d'une linéarisation exacte.

Mots-clés : écoulement multiphasique, système hétérogène, solution numérique, condition d'interface, solveur non-linéaire

Contents

List of Figures

List of Tables

Notation

Roman

B_i	control volume, box for vertex v_i
$\mathbf{d_k}$	NEWTON direction
$dtscale$	scaling factor for time step
e	element
E	set of elements
E_i	set of elements adjoined to vertex v_i
$\mathbf{f}$	right-hand side for Jacobian system
F	defect term
$\mathbf{F}$	equation system
$\mathbf{g}$	gravity vector $[m/s^2]$
$\mathbf{g_k}$	direction of the gradient
G	domain of interest
$k_{r\alpha}$	relative permeability of phase α $[-]$
K	intrinsic permeability scalar $[m^2]$
$\mathbf{K}$	intrinsic permeability tensor $[m^2]$, matrix for Jacobian system
$\mathbf{K_T}$	tangent element matrix
l	counter
n_{ls}	number of line search steps
n_{phas}	number of phases (here equal 2)
$\mathbf{n}$	normal vector for box face at integration point
N_i	shape function of vertex v_i
p	pressure $[Pa]$
p_c	capillary pressure $[Pa]$
p_d	displacement pressure, Brooks–Corey parameter $[Pa]$
p_e	entry pressure; capillary pressure of the biggest pore $[Pa]$
q	number of actual line search
q_α	source term for phase α $[m^3/(m^3 \cdot s]$, specific flux $[m/s]$

S	saturation $[-]$
S_e	effective saturation, $S_{we} = \frac{S_w - S_{wr}}{1 - S_{wr} - S_{nr}}$ $[-]$
S_r	residual saturation $[-]$
S_t	total liquid saturation $[-]$
t	time $[s]$
$\mathbf{u}$	vector of unknowns for Jacobian system, corrections for primary variables
v_i	vertex i
$\mathbf{v}$	Darcy velocity vector
V	set of vertices
W_i	weighting function for vertex v_i
x	spatial variable $[m]$
$\mathbf{x}$	vector holding the primary variables
$\mathbf{x}^k$	barycenter of element e_k
y	spatial variable $[m]$
z	geodetic height $[m]$

Greek

α	subscript indicating an arbitrary phase;
β	constant for the NEWTON method
δ_{ij}	Kronecker's delta (= 1 if $i = j$; = 0 else)
Δh	grid width $[m]$
Δt	numerical time step $[s]$
ϵ_{lin}	accuracy of the linear solution in the NEWTON method
ϵ_{nl}	accuracy of the nonlinear solution in the NEWTON method
$\gamma_{ij}^{FUe_l}$	integral over sub-control volume face between vertex v_i and vertex v_j in Element e_l
Γ	Boundary between two subdomains of G
Γ_{B_i}	integration path around control volume B_i
$\Gamma_{\alpha,D}$	Dirichlet boundary
$\Gamma_{\alpha,N}$	Neumann boundary

η	damping factor for the NEWTON method
λ	Brooks–Corey parameter [−]
λ_α	mobility of phase α $[(ms)/kg]$
μ	dynamic viscosity $[kg/(ms)]$
ρ	density $[kg/m^3]$
ϕ	porosity [−]
ψ	total potential [Pa]
Θ_k	angel between NEWTON direction and gradient

Subscripts

G_i	belonging to subdomain G_i; e.g. $x_{G_i} \in G_i$ or $p\|G_i$ for the pressure in subdomain G_i
i	index; counter
init	initial
j	index; counter
k	index; counter
l	index; counter
n	non-wetting phase or NAPL phase
t	total
w	wetting phase or water phase

Superscripts

k	time step
r	iteration step

Abbreviations

BC	Brooks–Corey
BOX	box method
DNAPL	non-aqueous phase liquid denser than water
FE	finite elements

FU	fully upwind
FV	finite volumes
FUB	fully upwind box method (fully upwind finite volume method)
ILU	incomplete decomposition solver
$Inv_{pc}^{G_i}$	inverse capillary pressure saturation relationship for subdomain G_i
NAPL	none-aqueous phase liquid
PPS	Phase Pressure Saturation (formulation); usually $p_w - S_n$ formulation
PPSIC	Phase Pressure Saturation (formulation) with Interface Condition; usually $p_w - S_n$ formulation with extended capillary pressure equilibrium condition
REV	Representative Elementary Volume
VG	van Genuchten

1 Introduction

Groundwater is an essential element of the hydrological cycle. It is the most important contribution to drinking water supply in many countries all around the world. Pollution of groundwater resources can therefore seriously endanger lives. Thus it is very important to determine flow and transport paths in the subsurface. The tools for prediction are numerical models.
The subsurface hydrosystem is a multiphase system consisting of groundwater, soil air and often also pollutants which are not soluble in water and thus form an extra phase (NAPLs: Non-Aqueous Phase Liquids). The flow and transport processes take place at different scales ranging from the microscale and the pore scale to the regional scale at which whole groundwater reservoirs can be considered.
On all scales strong spatial variations of material properties may occur. These have to be taken into account when modelling flow and transport phenomena in the subsurface. Macroscale heterogeneities, which will be the topic of this work, have to be taken into account very carefully as they have an enormous influence on the flow and transport behaviour. Besides describing the mere physical dimensions and properties of a block heterogeneity, the interface between the media with different properties has to be duely considered. A physically correct description of the entering and leaving of the different rocks has to be provided and moreover a good performance of the numerical scheme has to be ensured.
To represent the flow and transport processes between two different types of rocks in a physically correct manner, MOLENAAR and DE NEEF have developped an interface condition to ensure that the non-wetting phase cannot enter a less permeable material until the physical conditions for the entering (sufficient entry pressure) are reached, see [6]. This *interface condition* (or *transition condition* which is used synonymously with interface condition in this thesis) is explained in Section 3.5.
JAKOBS et al. investigated the performance of the numerical simulator MUFTE-ug using the BOX scheme with the interface condition given by MOLENAAR and DE NEEF [16]. Comparisons with computations carried out without using the interface condition showed, that the transition condition is efficient under certain conditions, but encounters problems under special circumstances (steep relative permeability - saturation relationships), see Figure 1. In this figure, k_{rw} represents the relative permeability of the wetting phase, k_{rn} that of the non-wetting phase. The numbers, 0.25, 1.0, and 4.0 on the labels on the relative permeability curves indicate the steepness of the respective function at saturation equal 1, which proved to be the crucial point for the performance of the scheme.
To cope with these difficulties and to improve the performance of the code with the transition condition the idea was to combine the interface condition with an "exact"

linearization of the partial differential equations (PDE's). The derivation of the global algorithm and matrices is described in HELMIG (1993) [11].
The aim of this Master's Thesis is

1. to correctly represent the processes at the interface between different rock types by using a special interface condition,
2. to combine this interface condition with a consistent linearization of the partial differential equations,
3. to implement it into the MUFTE code,
4. to compare the performances of the newly developped numerical scheme with those of existing numerical schemes (numerically linearized BOX scheme without and with the interface condition) and
5. to apply them to some theoretical and practical examples.

In Chapter 2 the fundamental definitions needed for the mathematical model developped in Chapter 3 are explained. Depending on this mathematical model the numerical model described in Chapter 4 is derived. Finally, the numerical scheme is applied to principle and realistic 1D and 2D examples and the performance of the new scheme is compared to the performances of other current schemes (Chapter 5). Proposing a possible amelioration stategy in Chapter 6 we conclude with some final remarks (Chapter 7).

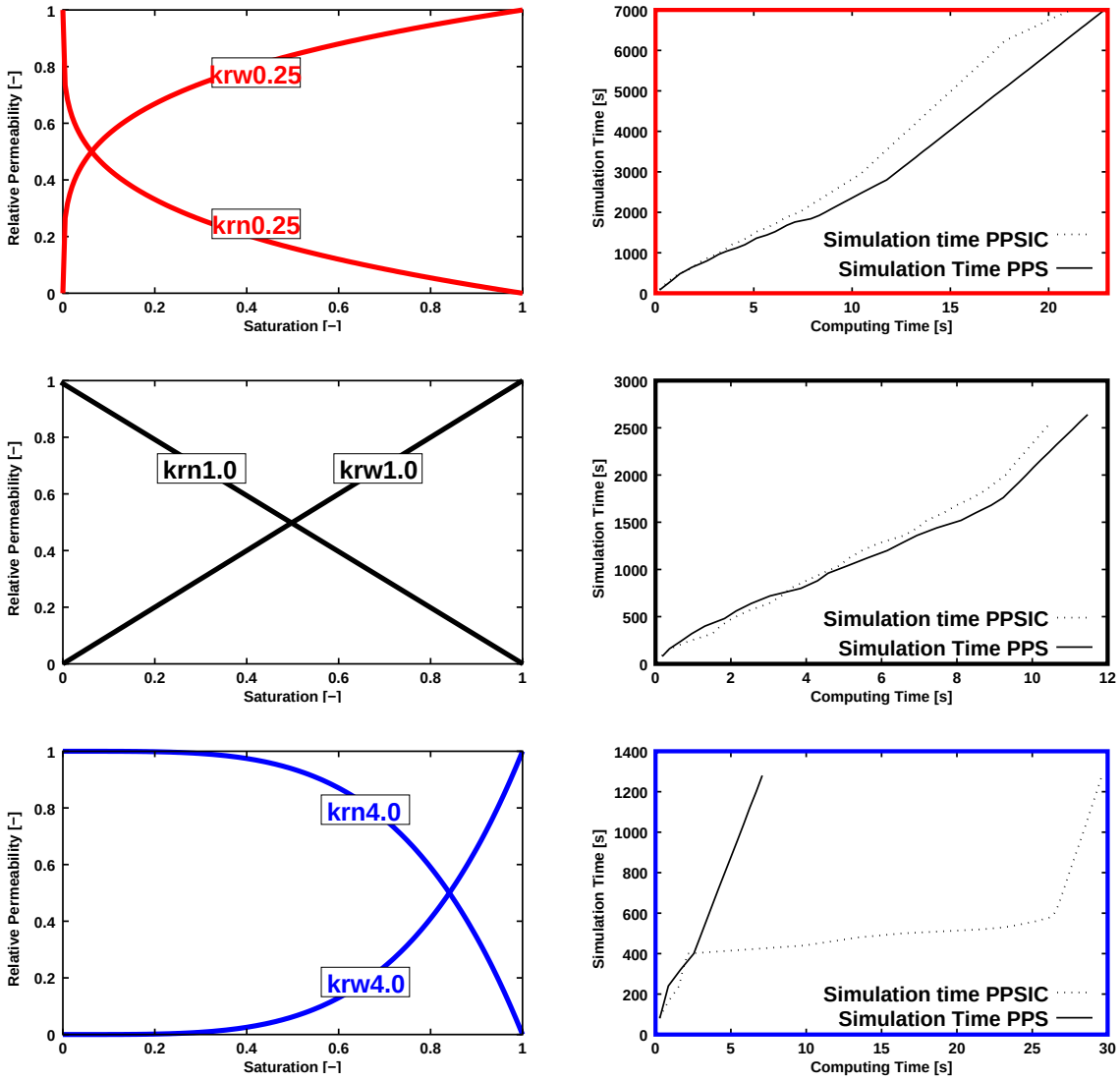

Figure 1: Performance of the code using the interface condition (PPSIC, phase pressure saturation formulation with interface condition) in comparison with the common method (PPS, phase pressure saturation formulation without interface condition). [15]

2 Fundamental Principles and Definitions

Basic definitions concerning scale and parameters are presented first. The selected scale determines which physical processes are taken into account. On the *molecular* scale single molecules are considered. These are decisive for fluid properties such as viscosity, density, and interfacial tensions. By averaging over a large number of molecules, we reach the *continuum* scale. The consideration of a continuum will be persued in the following. On the *micro* scale discontinuities at interfaces between different phases can be detected. Thus, at one point in space, exactly one phase is present. When talking about the *macro* scale, the micro scale properties are averaged over a *representative elementary volume* (REV). This leads to new parameters like porosity or saturation with new equations. If we consider one of these parameters, e.g. porosity, it becomes obvious, that if we choose a small averaging volume it might well lie either totally within solid rock or totally within the pore space, i.e. $\phi = 0$ or $\phi = 1$. Enlarging this volume we will encounter oscillations starting from extreme values and stabilizing at a more or less constant value, see Figure 2 left hand side, until larger scale heterogeneities are included into the averaging volume. The REV is the minimum volume for which the averaging parameters remain constant for the first time when further enlarging this volume, see Figure 2, right hand side (from [16]). The inclusion of larger scale heterogeneities leads to derivations from this constant value. The consideration of macro scale heterogeneities and thus the concept of the REV will be essential in this work.

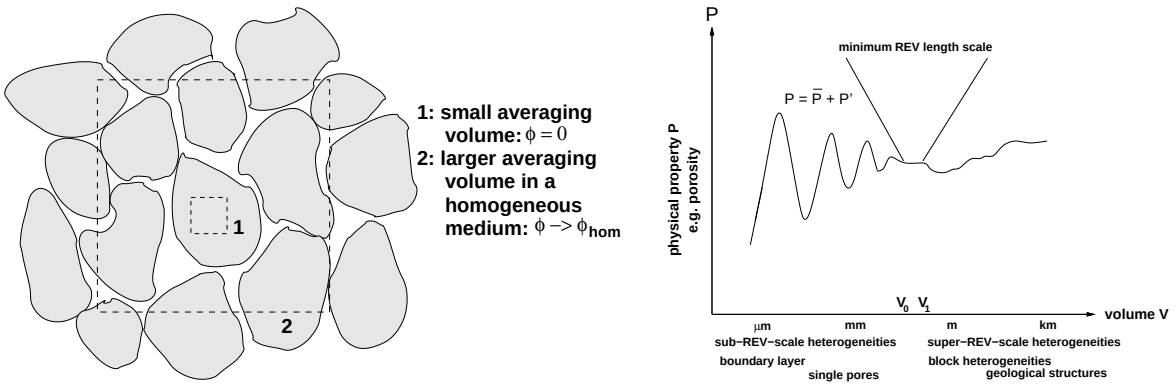

Figure 2: Definition of the representative elementary volume (REV)

Before discussing the mathematical model the relevant parameters for the macro scale have to be defined. The *saturation* of phase α is defined to be the ratio of the volume

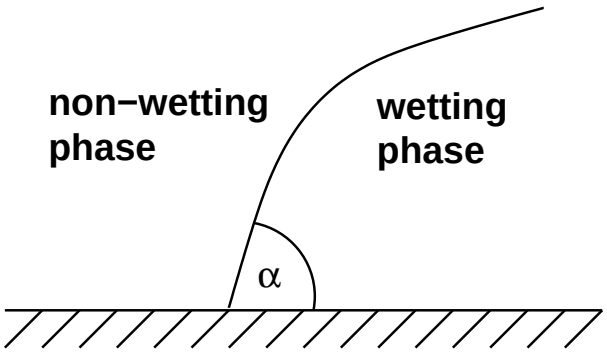

Figure 3: Definition of the the terms wetting and non-wetting fluid, respectively

of fluid α within the REV to the volume of the pore space within the REV:

$$S_\alpha(x,t) = \frac{\phi_\alpha(x,t)}{\phi}.$$

For saturation and phase *porosity* the following relationships are valid:

$$\begin{aligned} \sum_{\alpha=1}^{n_{phas}} S_\alpha &= 1 \qquad \text{for} \quad 0 \le S_\alpha \le 1, \\ \sum_{\alpha=1}^{n_{phas}} \phi_\alpha &= \phi \qquad \text{for} \quad 0 \le \phi_\alpha \le \phi, \end{aligned} \tag{1}$$

where n_{phas} is the number of phases in the model.
Another item to be discussed here is capillarity. Capillary effects are caused by cohesion within one phase and by adhesion between two different phases. These forces result in a wetting angle when two phases meet at an interface and a discontinuity of the pressure. The difference between the two meeting phase pressures is referred to as the *capillary pressure*

$$p_c = p_n - p_w, \tag{2}$$

where p_n is the pressure of the non-wetting and p_w the pressure of the wetting phase respectively. The fluid with a boundary angle $\alpha < 90°$ is referred to as the *wetting* fluid with respect to the solid phase, the fluid with an obtuse boundary angle $1 - \alpha > 90°$ is the *non-wetting* fluid, see Figure 3.
The microscale approach according to [12] shows that if a water-saturated soil is infiltrated by NAPL, the wetting phase retreats to smaller pores, as a decrease of meniscus radii corresponds to an increase in capillarity.

On the macro scale the capillary pressure is dependent on the saturation. If the saturation of the wetting fluid decreases it retreats to smaller and smaller pores which leads to an increase in capillary pressure. Thus, a macroscopic consideration of capillarity leads to a relationship of the form

$$p_c = p_c(S_w). \tag{3}$$

As a last parameter the relative permeabilities $k_{r\alpha}$ with $\alpha \in \{w, n\}$ have to be considered. On the micro scale the dependence of permeability on saturation can be easily explained. The presence of one fluid disturbs the flow of the other phase and vice versa. If the saturation of the wetting fluid decreases, not only the cross-sectional area available for the wetting phase flow decreases, but also the wetting fluid has to flow around those parts of the porous medium which are occupied by the non-wetting fluid. This results in increasing tortuosity of the flow paths for the wetting fluid.
Like for capillary pressure, the macroscale relative permeability is a function of the saturation

$$k_{r\alpha} = k_{r\alpha}(S_w). \tag{4}$$

The relations given in (3) and (4) depend on the material as well. They are determined by fitting the parameters of analytical functions to measured data, for more detail see Section 3.1. Our choice of the constitutive relationships, the functions in (3) and (4), is given in Subsections 3.1.1 and 3.1.2.

3 Mathematical Modelling of Two-Phase Flow

Like other physical quantities such as energy and momentum mass is conserved in nature. Conservation of mass for multiphase flow with respect to volume can be formulated as

$$\frac{\partial(S_\alpha \phi \rho_\alpha)}{\partial t} + \nabla \cdot (\rho_\alpha \mathbf{v}_\alpha) - \rho_\alpha q_\alpha = 0, \tag{5}$$

where ϕ is the porosity, S_α is the saturation of phase α, ρ_α the density, t time, $\mathbf{v}_\alpha$ is an average macroscopic pore velocity vector and q_α represents the source term.
The velocity vector $\mathbf{v}_\alpha$ is related to the gradient of the pressure p_α by the generalized Darcy law

$$\mathbf{v}_\alpha = -\frac{k_{r\alpha}}{\mu_\alpha}\mathbf{K}(\nabla p_\alpha - \rho_\alpha \mathbf{g}), \tag{6}$$

which is considered to be a momentum balance equation. $k_{r\alpha}$ represents the relative permeability, μ_α the dynamic viscosity, p_α the pressure of phase α, $\mathbf{K}$ the tensor of absolute permeabilities and $\mathbf{g}$ the vector of gravity. Inserting equation (6) into equation (5) leads to

$$\frac{\partial(S_\alpha \phi \rho_\alpha)}{\partial t} - \nabla \cdot (\rho_\alpha \frac{k_{r\alpha}}{\mu_\alpha}\mathbf{K}(\nabla p_\alpha - \rho_\alpha \mathbf{g})) - \rho_\alpha q_\alpha = 0. \tag{7}$$

While micro scale heterogeneities are included in Darcy's law macro scale heterogeneities have to be accounted for by the spatial variation of the parameters in Darcy's law (ϕ and $\mathbf{K}$) as well as by the variation of the parameters in the constitutive relationships, see section 3.1. For two phase flow, the multiphase flow equation (7) represents two equations, one for each of the two phases w and n. With the definition of the capillary pressure given in equation (2), p_c and $k_{r\alpha}$ still depend on the saturations of the phases, so that the closure relationship given in the first equation of (1) reduces to

$$S_w + S_n = 1 \tag{8}$$

and the following constitutive relationships given in (3) and (4) are needed

$$p_c = p_c(S_\alpha) \tag{9}$$

$$k_{r\alpha} = k_{r\alpha}(S_\alpha). \tag{10}$$

3.1 Constitutive Relationships

3.1.1 Capillary Pressure - Saturation Relationship

On the macro scale both capillary pressure and relative permeability depend on the saturation. If the water saturation decreases the wetting phase retreats into smaller and smaller pores and thus the capillary pressure increases. The macroscopic consideration of the capillarity results in a relationship given in equation (9).
Numerous parameterizations of the $p_c - S_w$ relationship exist. Among the most common ones are the relationships of BROOKS and COREY and of VAN GENUCHTEN. The principle difference between the two types of relationships is the behaviour of the function at $S_w = 1$. Whereas for VAN GENUCHTEN the capillary pressure is equal to zero for a fully water saturated soil, the BROOKS-COREY relation involves a displacement pressure (the pressure needed to displace the wetting phase from the largest pore, p_d). Even though measurements of capillary pressure for drainage processes (pressure cell technique) show, that p_c starts from 0 and not from a certain entry pressure, gamma-sorption techniques show, that such an entry pressure exists. As gamma-sorption is able to resolve interfaces, which is the main interest of this work (pressure cell techiques average over volumes), the BROOKS-COREY curve which directly implies the existence of an entry pressure is used.
The capillary pressure - saturation relationship according to BROOKS-COREY is defined with the help of the effective saturation S_e:

$$S_e(p_c) = \frac{S_w - S_{wr}}{1 - S_{wr}} = \left(\frac{p_d}{p_c}\right)^{\lambda} \quad \text{for } p_c \geq p_d. \tag{11}$$

The parameters

S_e	: effective saturation
S_{wr}	: residual water saturation
λ	: form parameter
p_d	: entry (displacement) pressure

characterize the pore space geometry. They are determined by fitting to measured $p_c - S_w$ curves.
The visualization of the BROOKS-COREY parameters can be found in Figure 4. As mentioned above, p_d is the displacement pressure. The parameter λ on the other hand describes the uniformity of the material. High λ parameters indicate a highly uniform material, usual values lie between 0.2 and 3.0.

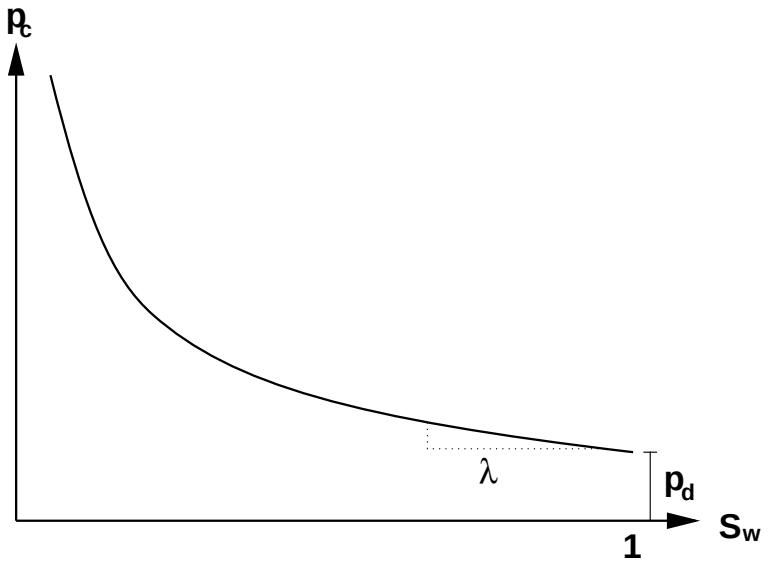

Figure 4: Typical BROOKS-COREY relation with visualization of parameters p_d and λ

3.1.2 Relative Permeability - Saturation Relationship

Like for the $p_c - S_w$ relationships macroscale $k_{r\alpha} - S_\alpha$ relationships can be evaluated by fitting to experimental data. In this work, we again use the BROOKS-COREY relationships - here in conjunction with the BURDINE theorem [4] - for the relative permeability curves:

$$k_{rw} = S_e^{\frac{2+3\lambda}{\lambda}} \tag{12}$$

$$k_{rn} = (1 - S_e)^2 \left(1 - S_e^{\frac{2+\lambda}{\lambda}}\right) \tag{13}$$

The form of the relative permeability-saturation relationships after BROOKS-COREY can be seen in Figure 5.

3.2 Selecting the Primary Variables

Different alternatives for choosing the primary variables can be distinguished: the choice depends on the problem to be solved.

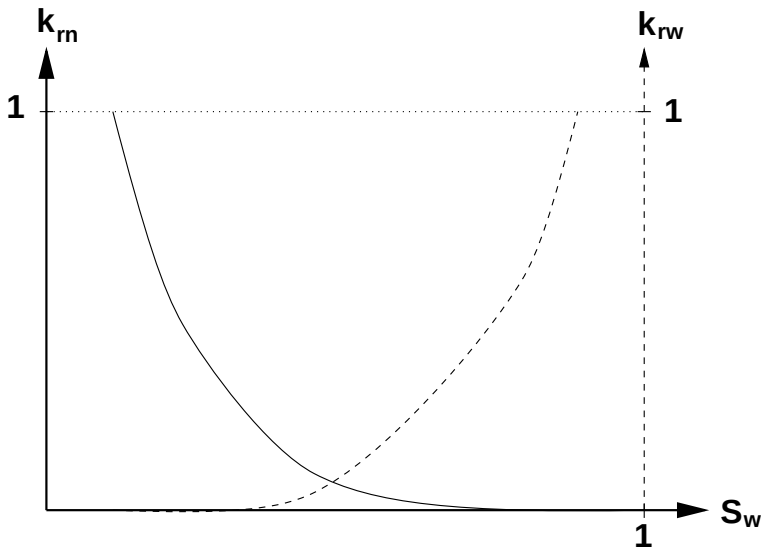

Figure 5: Relative permeability-saturation relationship after BROOKS-COREY

3.2.1 Pressure Formulation

The pressure formulation uses the phase pressures p_α as primary variables. Saturations are transformed to pressures by inverting the capillary pressure - saturation relationship. The constraint for using this formulation is, that the $p_c - S_w$ function has to be strictly monotonic in order to be invertable,

$$S_\alpha = g_\alpha(p_c).$$

The $p_w - p_n$ formulation reads as follows:

$$\begin{aligned} \text{wetting phase:} \quad & \frac{\partial(\phi\rho_w g_w)}{\partial t} - \nabla\cdot\left(\rho_w \frac{k_{rw}}{\mu_w}\mathbf{K}(\nabla p_w - \rho_w \mathbf{g})\right) - \rho_w q_w &= 0, \\ \text{non-wetting phase:} \quad & \frac{\partial(\phi\rho_n g_n)}{\partial t} - \nabla\cdot\left(\rho_n \frac{k_{rn}}{\mu_n}\mathbf{K}(\nabla p_n - \rho_n \mathbf{g})\right) - \rho_n q_n &= 0. \end{aligned} \tag{14}$$

These two equations represent a strongly coupled, nonlinear parabolic system

disadvantage: For capillary pressure gradients $\frac{\partial p_c}{\partial S_\alpha} = 0$ or close to zero, problems with inversion particularly in heterogeneous systems will occur. For this reason, the formulation is not very convenient for the problem at hand.

3.2.2 Saturation Formulation

This formulation is only applicable to incompressible fluids. The $S_w - S_n$ formulation: after calculating the total velocity $\mathbf{v_t} = \mathbf{v_w} + \mathbf{v_n}$, inserting one equation into the other and making further changes, there remains only one differential equation for the saturation [12].

3.2.3 Pressure-Saturation formulation

This formulation uses the pressure of one phase and the saturation of the other phase as primary variables. Differentiating equations (2) and (8), we obtain

$$\begin{aligned} \nabla p_n &= \nabla(p_w + p_c) && (15) \\ \frac{\partial S_w}{\partial t} &= \frac{\partial}{\partial t}(1 - S_n) = -\frac{\partial S_n}{\partial t}. && (16) \end{aligned}$$

Inserting (15) and (16) into the two-phase flow equations (7) leads to the $p_w - S_n$ formulation ($p_n - S_w, p_w - S_w, p_n - S_n$ accordingly):

$$\begin{aligned} \text{w:} \quad & -\frac{\partial(\phi\rho_w S_n)}{\partial t} - \nabla \cdot \left(\rho_w \frac{k_{rw}}{\mu_w} \mathbf{K}(\nabla p_w - \rho_w \mathbf{g}) \right) - \rho_w q_w = 0 \\ \text{n:} \quad & \frac{\partial(\phi\rho_n S_n)}{\partial t} - \nabla \cdot \left(\rho_n \frac{k_{rn}}{\mu_n} \mathbf{K}(\nabla p_w + \nabla p_c - \rho_n \mathbf{g}) \right) - \rho_n q_n = 0 \end{aligned} \qquad (17)$$

As with the pressure formulation, we also have a strongly coupled, nonlinear parabolic system consisting of two equations.

The advantages of this formulation are the following:

1. No problems with small capillary pressure gradients, as the capillary pressure is directly included in the equation system. Like this it is also applicable to heterogenous systems.

2. In the case of two immiscible fluids, decoupling of the equation system is possible (pressure + saturation equation) $\Longrightarrow$ IMPES method (implicit pressure – explicit saturation) [14].

As we are dealing with heterogenous systems a pressure-saturation formulation is chosen namely the $p_w - S_n$ formulation. In the following only this formulation will be considered.

Remark: Another possible formulation is the *global pressure formulation*. In order to have an equation similar to the single-phase flow equation, a non-physical variable, the

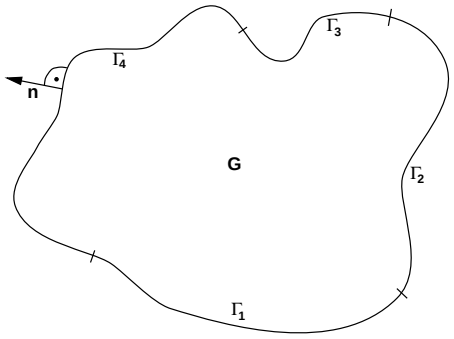

Figure 6: Definition of boundary conditions

so called "global pressure" is introduced. The two-phase flow equations are transferred to a coupled system of one parabolic saturation equation and one elliptic pressure equation. The two presssure unknowns p_w and p_n are replaced by one unknown, the *global pressure*, which is a functtion of p_w, p_n and S_e. For more details see [5].

3.3 Boundary and Initial Values

The two-phase flow equations represent a system of partial differential equations which describe a boundary and initial value problem. Thus the total boundary ∂G of the domain has to be described via boundary values. Moreover, initial values are needed for the whole domain G.
To be able to specify different sorts of boundaries, the boundary of the domain can be split up into different boundary segments, see figure 6:

$$\partial G = \Gamma_1 \cup \Gamma_2 \cup \ldots \cup \Gamma_n.$$

The two basically different types of boundaries are DIRICHLET and NEUMANN boundaries. If $\mathbf{u}$ denotes the vector of the unknowns (the primary variables, in our case p_w and S_n), DIRICHLET boundaries directly specify the value of $\mathbf{u}$ on the boundary. NEUMANN boundaries on the other hand specify the value of the derivative of $\mathbf{u}$ in the direction of the outer normal vector $\mathbf{n}$: $\frac{\partial u}{\partial n}$. Thus, this boundary condition is used to describe a flux across the boundary ∂G. A third type of boundary conditions exists (the CAUCHY boundary condition) which are constructed as a linear combination of the two previously mentioned boundary conditions.
The initial values finally define values for p_w and S_n for the initial state in the whole domain, i.e. $t = 0$.

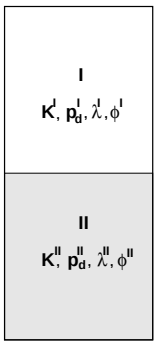

Figure 7: Discontinuity of parameters across the interface

3.4 Inclusion of Heterogeneity

For modelling macro scale heterogeneities an abrupt changing of rock properties such as porosity ϕ or absolute permeability $\mathbf{K}$ is a good approximation for realistic situations, see Figure 7. Thus, the respective parameters are assumed to be discontinuous across the interface between two different rock types.
The most difficult task is to determine discontinuous primary variables across the interface. To solve the flow problem, two transition conditions are needed at the interface [12]:

- **continuity of flux**
  The flux of both phases across the interface has to be continuous:

$$q_{\alpha I} = q_{\alpha II}$$

- **continuity of intensive state variables**
  The capillary pressure is continuous at the interface:

$$p_c^I = p_c^{II}.$$

As can be derived from the capillary pressure - saturation curve given in Figure 8, continuity in capillary pressure ($p_c^I = p_c^{II}$) involves a discontinuity in saturation ($S_w^I \neq S_w^{II}$) which should also be taken into account by the model. This is done by implementation of the interface condition, see Section 3.5.

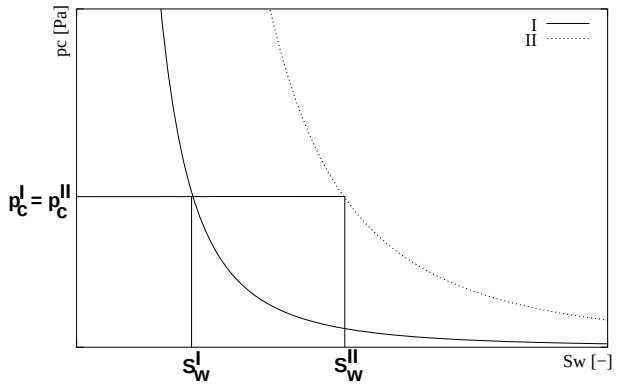

Figure 8: Continuity of capillary pressure, discontinuity of saturation across the interface

3.5 The Interface Condition

As mentioned in Section 3.4, discontinuous materials are a good approximation for heterogeneities in macroscale models. To maintain the macroscale momentum balance, i.e. the extended Darcy law given in (6), the capillary pressure p_c has to be continuous at the interface between different materials. This requirement, however, implies the discontinuity of saturation at the interface, see Figure 9. As the saturations are defined at the nodes of the FE mesh, a discontinuity of saturation can only be put into practice if these nodes lie directly on the interface. Using the box method, the interface Γ between two different subdomains with different properties normally coincides with the one between control volumes (patch oriented approach). For the implementation of the interface condition however, the grid is shifted so that the interface coincides with the FE grid, see Figure 10 (element oriented approach). Whereas for the box method without interface condition, non-wetting phase can enter the material with the lower entry pressure before the entry pressure p_d of the respective material is reached, this is no longer possible when implementing the interface condition. For this reason, the box method with interface condition, rather the box method without interface conditon, represents the interface in a physically correct manner.
Using the interface condition, the capillary pressure p_c becomes a quasi primary variable at the interface between two materials. If G_2 is a subdomain with a higher entry pressure than a neighboring subdomain G_1, the wetting-phase saturation in subdomain G_2 at the interface is equal to [16]:

$$S_w|G_2^\Gamma = \begin{cases} 1 & \text{if} \quad S_w|G_1^\Gamma \geq S_w^* \\ [p_c]_{G2}^{-1} & \text{if} \quad S_w|G_1^\Gamma < S_w^* \end{cases} \tag{18}$$

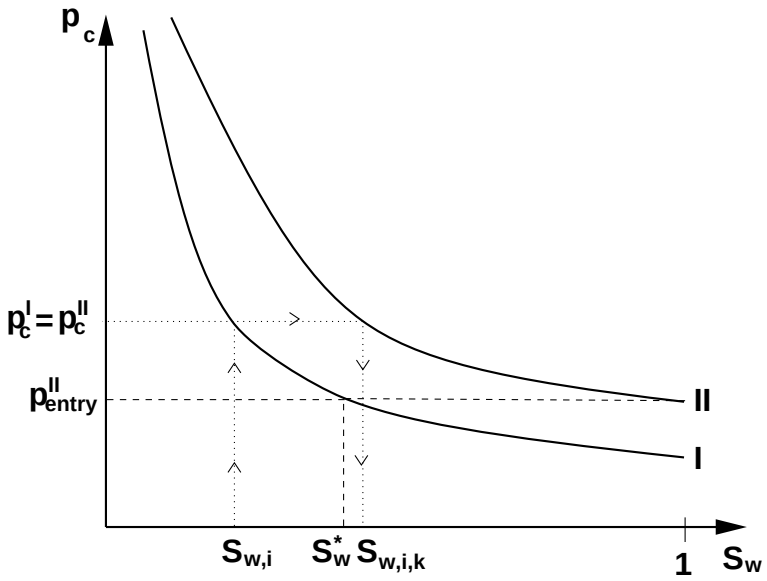

Figure 9: Continuity of capillary pressure, discontinuity of saturation at the interface

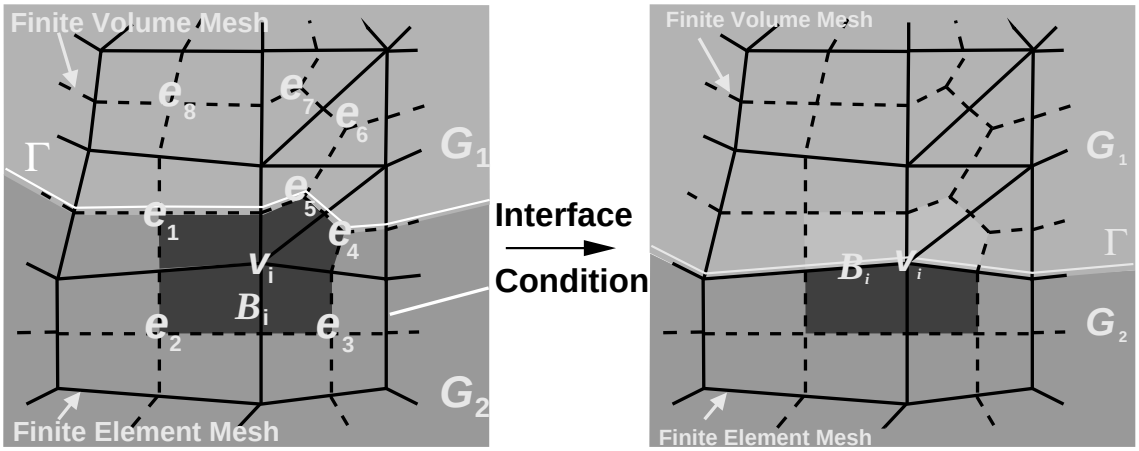

Figure 10: Patchoriented (left) and elementoriented grid (right hand side) [16]

where $[p_c]_{G2}^{-1}$ is the inverse of the $p_c - S_w$ function in G_2 and S_w^* is the saturation in G_1 for which the entry pressure of G_2 is reached. Moreover, $S_w|G_1^\Gamma$ is the wetting-phase saturation in subdomain G_1 at the interface, $S_w|G_2^\Gamma$ the one in G_2 at the interface, respectively.
To implement the transition condition (18) in the box scheme, we associate to each node v_i of the FE mesh a minimum capillary pressure $p_{c,min}^i$ which is simply $p_c(S_w, i)$ if v_i is not on the interface but is in the interior of one of the domains G_j. Otherwise $p_{c,min}^i$ is the minimum over all domains G_j having v_i on its boundary of $[p_c]_{G_j}(S_{w,i})$:

$$p_{c,min}^i = \min_{k \in E(i)} p_c(x^k, S_{w,i})$$

where $E(i)$ is the set of indices of the elements which have v_i as a corner, x^k is the barycenter of element e_k, and $p_c(x^k, \cdot)$ is the capillary pressure function $[p_c]_{G_j}$ for which $x^k \in G_j$.
We shall also define for each node v_i and each element e_k having v_i as a vertex a "virtual" saturation $S_{n,i,k}$, virtual because it does not appear in the output files but is only used for the elementwise computation of the entries of the local stiffness matrices. If v_i is not on an interface, $S_{n,i,k} = S_{n,i}$ for each element having v_i as a vertex. If v_i is on the interface and e_k is a subdomain having v_i as a vertex there are three possibilities:

- e_k is contained in the domain G_j at which the minimum capillary pressure $p_{c,min}^i$ for $S_{w,i} = 1 - S_{n,i}$ is realized, i.e. in general the material with the lowest entry pressure. In this case $S_{n,i,k} = S_{n,i}$.
- e_k is contained in a domain G_j whose entry pressure $[p_c]_{G_j}(1)$ is larger than the minimum capillary pressure $p_{c,min}^i$. In this case, $S_{n,i,k}$ is set to zero.
- e_k is not contained in the domain with the lowest entry presure, but $p_{c,min}^i$ is greater than the entry pressure of the domain G_j including e_k. Then we let $\tilde{S}_{w,i}$ be the wetting phase saturation for which $[p_c]_{G_j}(\tilde{S}_{w,i}) = p_{c,min}^i$; i.e. $\tilde{S}_{w,i} = [p_c]_{G_j}(p_{c,min}^i)$. Then we put $S_{n,i,k} = 1 - \tilde{S}_{w,i}$.

This can be described in a general formulation:

$$S_{n,i,k} = \begin{cases} S_{n,i} & \text{if } p_c(x^k, 1 - S_{n,i}) = p_{c,min}^i \\ 0 & \text{if } p_{c,min}^i < p_c(x^k, 1) \\ 1 - S_{w,i} & \text{where } S_{w,i} \text{ solves } p_c(x^k, S_{w,i}) = p_{c,min}^i. \end{cases} \tag{19}$$

The implementation of the interface condition and the computation of the minimum capillary pressure is explained in detail in Subsections 4.4.2 and 4.4.3.

However, there is a disadvantage to using the interface condition: due to the evaluation of the constitutive relationships with two different saturations at one node at the interface, mass conservation is no longer guaranteed.

3.6 Dealing with Nonlinearities

Numerically two different ways of dealing with nonlinearities can be distinguished. Which method is actually appplied, depends on the intensity of the nonlinearity.

3.6.1 Weak Nonlinearities

Weak nonlinearities can simply be taken into account using linear interpolation. Variables, which are a function of one of the phase pressures can be regarded as being weakly nonlinear in our case, see *Helmig*, 1993 [12], Chapter 2.

3.6.2 Strong Nonlinearities

To deal with strong nonlinearities, a consistent linearization is used in order to linearize the system, see Section 4.4. Variables depending on saturations can be regarded as being strongly nonlinear. Thus, the $p_c - S_w$ as well as the $k_{r\alpha} - S_\alpha$ relationships definitely belong to this class of strongly nonlinear functions, see again *Helmig*, 1993 [12], Chapter 2.

4 Numerical Model

As the multiphase flow equations given in equation (17), which represents a system of nonlinear coupled differential equations, generally cannot be solved analytically, a numerical model is usually necessary.

4.1 Discretization

Discretization methods are used in order to replace the differential terms in the two phase flow equations by algebraic terms. The requirements for convergence of the discretization methods are, that they must be consistent and stable. Moreover it is desirable to have a monotonic solution behaviour (which means in our case a non-oscillating solution) in order to avoid over- or undershooting. To prevent the appearance of non-physical sinks and sources, a conservative discretization is to be persued. In order to combine the advantages of the (cell-centered) finite volume (FV) discretization and the finite element (FE) discretization, the BOX scheme was developped. This discretization technique is not only conservative (like the FV method), but can also be applied to unstructured grids easily (like the FE method).

4.2 Subdomain Collocation Finite Volume Method (BOX method)

For the box method [1],[16], two different grids are needed. Boxes (control volumes) are constructed around the nodes of the initial FE mesh, which defines the elements. This construction is done by linking the barycenter of each element with the midpoints of the edges of this element, see Figure 11. We define a weighting function equal to 1 inside a box, B_i and equal to 0 outside the box:

$$W_i(x) = \begin{cases} 1 & \text{if} \quad x \in B_i \\ 0 & \text{if} \quad x \notin B_i. \end{cases} \tag{20}$$

The dimension of this space is equal to the number of nodes N of the FE mesh, and the basis function (or shape function) associated with the i^{th} vertex is N_i which is equal to 1 at node i of the FE mesh and equal to zero at all other nodes:

$$N_i = \delta_{ij}, \tag{21}$$

with δ_{ij} the Kronecker delta. Both p_w and S_n will be approximated in the space of piecewise polynomial functions which are first order (affine linear) on the triangles of the FE mesh and which are first order in each variable separately (affine bilinear) on each rectangle of the FE mesh.

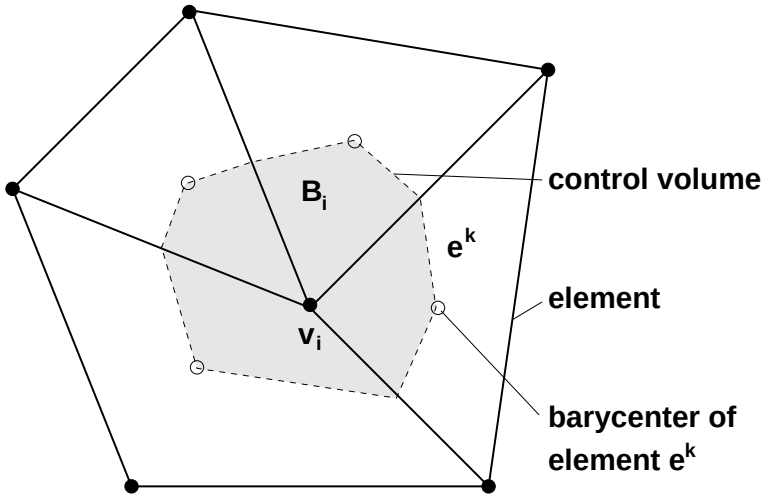

Figure 11: Construction of a control volume

For time discretization, we use a fully implicit Eulerian approach, see Section 4.5. The matrix of the accumulation term is approximated with a *lumped mass matrix* M_{ij}^{lump}

$$M_{ij}^{lump} = \delta_{ij} \sum_k M_{ik}. \tag{22}$$

with

$$M_{ij} = \int_B W_i N_j dB \tag{23}$$

as the *consistent mass matrix*.
With the definition of the mobilities $\lambda_\alpha = \frac{k_{r\alpha}}{\mu_\alpha}$ and the assumptions of time independence of the porosity ϕ and incompressibility, the two-phase flow equations (7) become

$$\frac{\partial S_\alpha}{\partial t}\phi - \nabla \cdot (\lambda_\alpha \mathbf{K} (\nabla p_\alpha - \rho_\alpha \mathbf{g})) - q_\alpha = 0. \tag{24}$$

Multiplying by our weighting function W_i and integrating (24) over the whole domain G results in integrals over the boxes:

$$\int_{B_i} \frac{\partial S_\alpha}{\partial t}\phi W_i dB_i - \int_{B_i} \nabla \cdot (\lambda_\alpha \mathbf{K} (\nabla p_\alpha - \rho_\alpha \mathbf{g})) W_i dB_i - \int_{B_i} q_\alpha W_i dB_i = 0. \tag{25}$$

We approximate the unknowns $p_{w,i}$ and $S_{n,i}$ as well as the auxiliary function $p_{c,i}$ and the gravity function g_i by shape functions N_j in the following way:

$$u \approx \sum_{j \in V} u_j N_j, \tag{26}$$

where V represents the set of vertices of the FE mesh. The gradients of these functions at node i are approximated as

$$\nabla \sum_j u_j N_j = \sum_{j \neq i} (u_j - u_i) \nabla N_j. \tag{27}$$

Applying the GAUSS integral theorem and using an element oriented approach which is necessary for the implementation of the interface condition, see Section 3.5, yields the following form of the two-phase flow equations, discretized in time and space:

$$\begin{aligned} f_{\alpha i}(S_{ni}^{k+1}; S_{ni}^{k}; p_{wi}^{k+1}; p_{wj}^{k+1}) \quad := \quad & \\ & - \sum_{k \in E(i)} \frac{\phi|_{e^k}(x_i)}{\Delta t} (S_{n,i,k}^{k+1} - S_{n,i,k}^{k}) \operatorname{meas} b_{i,k} \\ & - \sum_{k,j} \lambda_{\alpha,ij,k}^{k+1} \mathbf{K}(x_k) \left[\sum_{m \in V} p_{\alpha,m}^{k+1} \, \mathbf{grad} \, N_m(x_{ij,k}) \right. \\ & \left. - \rho_\alpha N_m(x_{ij,k}) \mathbf{g} \right] n_{ij,k} \operatorname{meas}(\gamma_{ij,k}) \\ & - q_{\alpha i}^{k+1} \operatorname{meas} B_i - m_{\alpha i} \\ & = 0, \end{aligned} \tag{28}$$

with $n_{ij,k}$ the outer normal vector, $\mid B_i \mid$ represents the area (2D), respectively the volume (3D), of the control volume around node i, $b_{i,k}$ the subcontrol volume of B_i which corresponds with element k, the upper indices k as well as $k+1$ denote the time step, and j represents a neighboring node of i. The integrals over the boundaries of B_i, Γ_{B_i}, are evaluated by using the midpoint rule; i.e., the integral over a segment of the boundary is calculated in multiplying the value at the midpoint of the control volume boundary segment by its length $\gamma_{ij,k}$.
Defining

$$\psi_{\alpha i}^{k+1} := p_{\alpha,i}^{k+1} - \rho_\alpha g z_i,$$

the sign of $\psi_{\alpha j} - \psi_{\alpha i}$ gives the direction of the flow of phase α. The parameter z_i is the geodetic height of vertex v_i and $m_{\alpha i}$ is the flow over $\partial B_i \cap \Gamma_N$ where Γ_N is a NEUMANN type boundary.

Using the fully upwind (FU) finite volume method for the mobilities results in

$$\lambda_{\alpha ij}^{FU_{e_l}} = \begin{cases} \lambda_{\alpha j} & \text{if} \quad (\psi_{\alpha j} - \psi_{\alpha i}) \geq 0 \\ \lambda_{\alpha i} & \text{if} \quad (\psi_{\alpha j} - \psi_{\alpha i}) < 0 \end{cases}, \tag{29}$$

which means that the mobility of the node with the higher potential is chosen.
The evaluation of quantities depending on saturation is done in the following way:

$$p_c|_{e_k}(x) = \sum_{m \in V(k)} p_c(x^k, 1 - S_{n,m,k}) N_m(x) \tag{30}$$

$$p_n|_{e_k}(x) = \sum_{m \in V(k)} \Big(p_{w,m} + p_c(x^k, 1 - S_{n,m,k}) N_m(x) \Big) \tag{31}$$

$$k_{rw}|_{e_k}(x) = \sum_{m \in V(k)} k_{rw}(x^k, 1 - S_{n,m,k}) N_m(x) \tag{32}$$

$$k_{rn}|_{e_k}(x) = \sum_{m \in V(k)} k_{rn}(x^k, S_{n,m,k}) N_m(x) \tag{33}$$

The box method is locally mass conservative, as exactly the same term occurs for the boundary integral of two neighboring control volumes.
If the mobilities were weighted between two neighboring nodes this would lead to a non-physical flux from the domain of lower to the domain of higher permeability (e.g. central weighting, harmonical weighting). This non-physical flux however can be avoided using Fully Upwinding (Figure 12). Although the Fully-Upwind method gives better results then for example the Standard or Petrov-Galerkin methods [13], there is still an error depending on the grid width. Using Fully-Upwinding, it is however impossible to reproduce the discontinuity of saturation at the interface between different materials according to Section 3.4. To avoid this error, the interface condition is to be implemented.

4.3 Consistent Linearization

The two-phase flow equations (7) are of the form

$$\mathbf{f}(\mathbf{u}) = \mathbf{K}(\mathbf{u}) \cdot \mathbf{u} - \mathbf{r}(\mathbf{u}) = 0 \tag{34}$$

where $\mathbf{u}$ is the vector of the unknowns, in our case p_w and S_n. The functional $\mathbf{f}(\mathbf{u})$ is strongly nonlinear as the constitutive relationships ($p_c - S_\alpha$, $k_{r\alpha} - S_\alpha$) are strongly nonlinear.

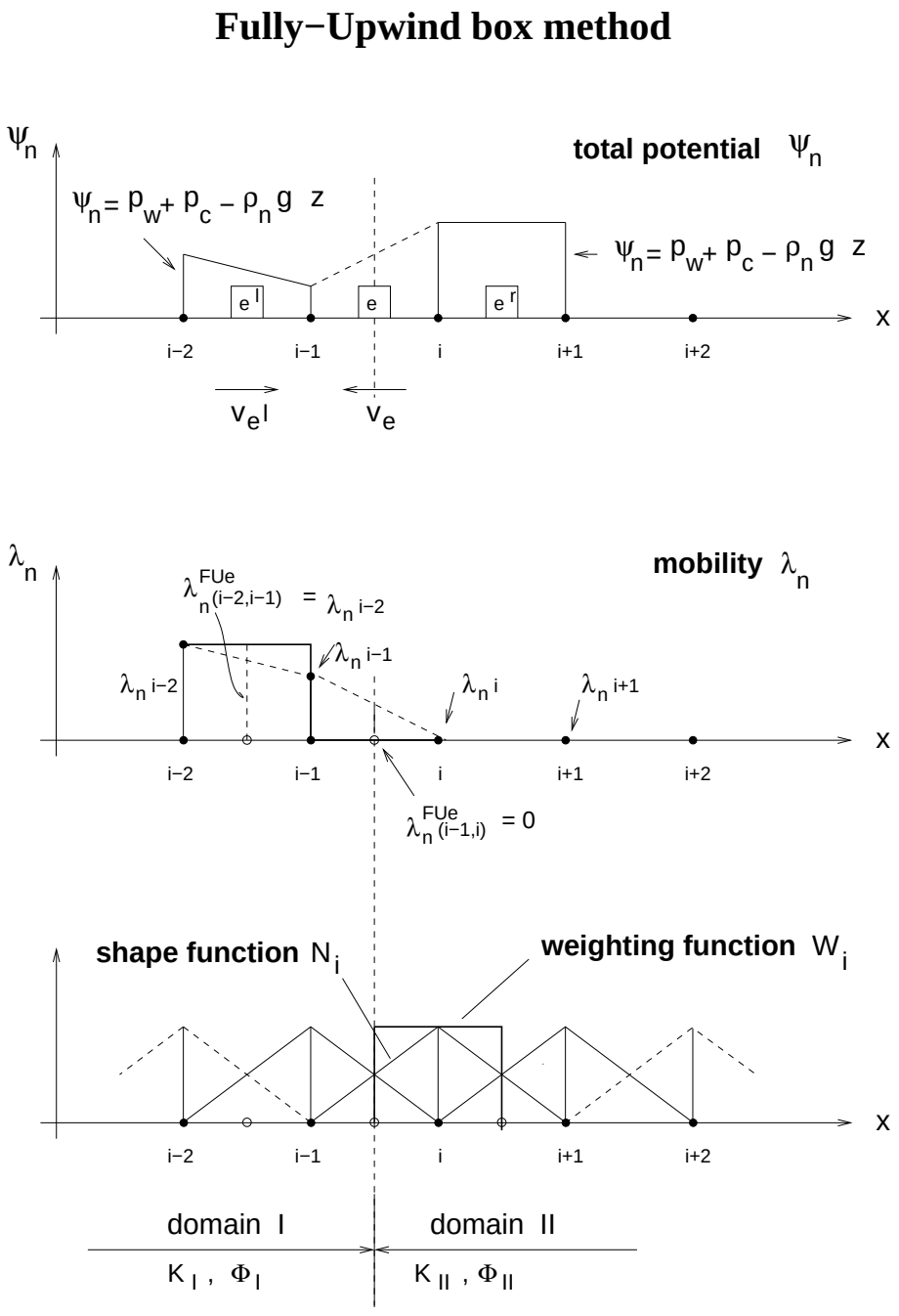

Figure 12: BOX method using Fully Upwinding [16]

To linearize the equation, a NEWTON-RAPHSON linearization is used which has the advantage of having a quadratic convergence behaviour, compared for example to the PICARD linearization which converges only linearly. Using the NEWTON linearization, the nonlinear functional $\mathbf{f}$ is expanded into a Taylor series where the terms of higher order are neglected [12]:

$$\mathbf{u}^{n+1,r+1} = \mathbf{u}^{n+1,r} - \underbrace{\left(\frac{\partial \mathbf{f}}{\partial \mathbf{u}}\right)^{-1}_{n+1,r}}_{\text{inverse of Jacobian of } r^{th} \text{ iteration}} \cdot \mathbf{f}(\mathbf{u}^{n+1,r}) \tag{35}$$

Usually, the entries of the Jacobian are differentiated *numerically* according to

$$J_{ij} = \frac{\partial f_i^{n+1,r}}{\partial u_j^{n+1,r}} \approx \frac{f_i(\ldots, u_{j-1}, u_j + \Delta u_j, u_{j+1}, \ldots) - f_i(\ldots, u_{j-1}, u_j, u_{j+1}, \ldots)}{\Delta u_j},$$

where $\Delta u_j = u_j \cdot \delta$ with δ a small increment.
In the following an "exact" calculation of the Jacobian entries $\left(\frac{\partial \mathbf{f}}{\partial \mathbf{u}}\right)$ is applied, namely the *consistent* or *exact linearization* [11].
The derivatives to be computed are the derivations of the functions $k_{r\alpha}$, ρ_α and p_c for the primary variables p_w and S_n. In our example, the two fluids are assumed to be incompressible which means that $\frac{\partial \rho_\alpha}{\partial S_\alpha} = 0$ and $\frac{\partial \rho_\alpha}{\partial p_\alpha} = 0$. Moreover, we assume $\frac{\partial k_{r\alpha}}{\partial p_w} = \frac{\partial p_c}{\partial p_w} = 0$ so that it is sufficient to calculate $\frac{\partial k_{r\alpha}}{\partial S_n}$ and $\frac{\partial p_c}{\partial S_n}$ as the only non-zero terms.
From equations (34) and (35) we obtain for an element e

$$\underbrace{\mathbf{K}_{T_{\alpha e}}^{n+1,r;n,0}(\mathbf{u})}_{I} \cdot \underbrace{\Delta \mathbf{u}_e^{n+1,r+1;n+1,r}}_{II} = \underbrace{\mathbf{K}_{E_{\alpha e}}^{n+1,r}(\mathbf{u}) \cdot \mathbf{u}_e^{n+1,r}}_{III} - \underbrace{\mathbf{r}_{\alpha e}^{n+1,r;n,0;n+1,0}}_{IV} \tag{36}$$

with

$$\begin{aligned} \mathbf{f}_{\alpha e}^{n+1,r;n,0} &= \mathbf{K}_{E_{\alpha e}}^{n+1,r} \cdot \mathbf{u}_e^{n+1,r} - \mathbf{r}_{\alpha e}^{n+1,r;n,0;n+1,0}, \\ \Delta \mathbf{u}_e^{n+1,r+1;n+1,r} &= \mathbf{u}_e^{k+1,r+1} - \mathbf{u}_e^{k+1,r}, \end{aligned}$$

and

$$\mathbf{K}_{T_{\alpha e}}^{n+1,r;n,0} = \frac{\partial \mathbf{f}_{\alpha e}^{n+1,r;n,0}}{\partial \mathbf{u}_e^{n+1,r}}.$$

The matrix $\mathbf{K}_{E_{\alpha e}}^{n+1,r}$ contains all the vectors which are differentiated at the new time step $k+1$ and the last known iteration step r, whereas $\mathbf{r}_{\alpha e}^{n+1,r;n,0;n+1,0}$ represents the known

parameters, i.e. the derivatives evaluated at the old time and iteration step k and respectively r. The evaluation of the terms I through IV can be found in appendix C. The analytical functionals $\mathbf{f}_\alpha(\mathbf{u})$ (see equation (24)), the discretization of these functionals in the BOX scheme and their derivatives at a vertex i are the following:

$$\begin{aligned} f_w(\mathbf{u}) &= -\phi\frac{\partial S_n}{\partial t} - \text{div}\,[\lambda_w \mathbf{K}(\,\mathbf{grad}\, p_w - \rho_w \mathbf{g})] - q_w = 0 \\ f_n(\mathbf{u}) &= \phi\frac{\partial S_n}{\partial t} - \text{div}\,[\lambda_n \mathbf{K}(\,\mathbf{grad}\, p_w + \mathbf{grad}\, p_c - \rho_w \mathbf{g})] - q_n = 0 \end{aligned} \tag{37}$$

$$\begin{aligned} f_{wi}(\mathbf{u}) &= -\phi\frac{S_{ni}^{k+1} - S_{ni}^{k}}{t^{k+1} - t^k} \mid B_i \mid - \int\limits_{\partial B_i} \lambda_w^{k+1} \mathbf{K}(\,\mathbf{grad}\,(p_w)_i^{k+1} - \rho_w \mathbf{g_i}) \cdot \mathbf{n} d\Gamma_{B_i} \\ &\quad -q_{wi}^{k+1} \mid B_i \mid = 0 \\ f_{ni}(\mathbf{u}) &= \phi\frac{S_{ni}^{k+1} - S_{ni}^{k}}{t^{k+1} - t^k} \mid B_i \mid - \int\limits_{\partial B_i} \lambda_n^{k+1} \mathbf{K}(\,\mathbf{grad}\,(p_w)_i^{k+1} + \mathbf{grad}\,(p_c)_i^{k+1} - \rho_w \mathbf{g_i}) \cdot \mathbf{n} d\Gamma_{B_i} \\ &\quad -q_{ni}^{k+1} \mid B_i \mid = 0 \end{aligned} \tag{38}$$

$$\begin{aligned} \frac{\partial f_{wi}}{\partial S_n} &= -\phi\frac{\mid B_i \mid}{t^{k+1} - t^k} - \int\limits_{\partial B_i} \frac{\partial \lambda_w^{k+1}}{\partial S_n} \mathbf{K}(\,\mathbf{grad}\,(p_w)_i^{k+1} - \rho_w \mathbf{g_i}) \cdot \mathbf{n} d\Gamma_{B_i} \\ \frac{\partial f_{wi}}{\partial p_w} &= -\int\limits_{\partial B_i} \lambda_w^{k+1} \mathbf{K}\frac{\partial}{\partial p_w}(\,\mathbf{grad}\,(p_w)_i^{k+1}) \cdot \mathbf{n} d\Gamma_{B_i} \\ \frac{\partial f_{ni}}{\partial S_n} &= \phi\frac{\mid B_i \mid}{t^{k+1} - t^k} - \int\limits_{\partial B_i} \frac{\partial \lambda_n^{k+1}}{\partial S_n} \mathbf{K}(\,\mathbf{grad}\,(p_w)_i^{k+1} + \mathbf{grad}\,(p_c)_i^{k+1} - \rho_w \mathbf{g_i}) \\ &\quad +\lambda_n^{k+1} \mathbf{K}(\frac{\partial}{\partial S_n}(\,\mathbf{grad}\,(p_c)_i^{k+1})) \cdot \mathbf{n} d\Gamma_{B_i} \\ \frac{\partial f_{ni}}{\partial p_w} &= -\int\limits_{\partial B_i} \lambda_n^{k+1} \mathbf{K}\frac{\partial}{\partial p_w}(\,\mathbf{grad}\,(p_w)_i^{k+1}) \cdot \mathbf{n} d\Gamma_{B_i} \end{aligned} \tag{39}$$

4.4 Implementation of the Transition Condition with Exact Linearization into the MUFTE code

When combining the interface condition and consistent linearization, it is not enough to implement the two methods separately at the same time into the MUFTE code. Due to their combination, new conditions evolve which do not exist either for the interface condition alone or for the consistent linearization alone.
Considering the saturation at the interface, it is obvious that it has to be equal to zero within a less permeable material until the entry pressure for this rock type is reached. This means however, that the differentiation of the storage term in the derivatives $\frac{\partial f_\alpha}{\partial S_n}$

is not to be calculated by differentiating the PDE's but by computing the derivatives in a way, that the respective derivatives of the storage term have to be set to zero as long as the entry pressure is not yet reached.

The discretized storage terms according to equation (38) are:

$$\begin{aligned} f_{wi}(\mathbf{u}) &= -\phi \frac{S_{ni}^{k+1} - S_{ni}^{k}}{t^{k+1} - t^{k}} \mid B_i \mid - \ldots \\ f_{ni}(\mathbf{u}) &= \phi \frac{S_{ni}^{k+1} - S_{ni}^{k}}{t^{k+1} - t^{k}} \mid B_i \mid - \ldots \end{aligned} \tag{40}$$

Mathematical differentiation of the storage term as for the exactly linearized box method without interface condition yields:

$$\begin{aligned} \frac{\partial f_{wi}}{\partial S_n} &= -\phi \frac{\mid B_i \mid}{t^{k+1} - t^{k}} - \ldots \\ \frac{\partial f_{ni}}{\partial S_n} &= \phi \frac{\mid B_i \mid}{t^{k+1} - t^{k}} - \ldots \end{aligned} \tag{41}$$

The state derivatives used for the exactly linearized box method with interface condition are:

$$\begin{aligned} &\text{if } p_{c,min} < p_{entry} && \partial \left(-\phi \frac{S_{ni}^{k+1} - S_{ni}^{k}}{t^{k+1} - t^{k}} \mid B_i \mid \right) / \partial \left(S_n \right) &&= 0 \\ &\text{else} && \partial \left(-\phi \frac{S_{ni}^{k+1} - S_{ni}^{k}}{t^{k+1} - t^{k}} \mid B_i \mid \right) / \partial \left(S_n \right) &&= \phi \frac{\mid B_i \mid}{t^{k+1} - t^{k}} \\ &\text{if } p_{c,min} < p_{entry} && \partial \left(\phi \frac{S_{ni}^{k+1} - S_{ni}^{k}}{t^{k+1} - t^{k}} \mid B_i \mid \right) / \partial \left(S_n \right) &&= 0 \\ &\text{else} && \partial \left(\phi \frac{S_{ni}^{k+1} - S_{ni}^{k}}{t^{k+1} - t^{k}} \mid B_i \mid \right) / \partial \left(S_n \right) &&= \phi \frac{\mid B_i \mid}{t^{k+1} - t^{k}} \end{aligned} \tag{42}$$

In the following, the implementation of the transition condition with exact linearization of the PDE's into the MUFTE code will be shown. For simplicity's sake, only the parts the most relevant for the implementation will be presented below. First, an overview of the implementation is given, then the functions and the meaning of some notation used in order to clarify the total overview of the implementation, has to be made clear. In the end, the hierarchy of the respective functions in the code is shown.

4.4.1 Total Overview of the Implementation

The output from the last time/ iteration step is always a value of the saturation S_n and of the pressure p_w. In order to restrict the wetting phase saturation to the interval $[0, 1]$ and to avoid extremely steep gradients, especially of the capillary pressure - saturation relationship, limiting conditions for S_n are introduced first. The relevant parts for the implementation of the interface condition with exact linearization are the following: first, the minimum capillary pressures with respect to the non-virtual saturation ($p_{c,min}$) and to an incremented non-virtual saturation ($p_{c,min,inc}$) are calculated. This is done in the function called `PreCapillaryPressure` in the MUFTE code, see Subsection 4.4.2. Next, the virtual saturations corresponding to S_n and to $S_{n,inc}$ are computed in the routine `TAssembleMatrixExact` in the MUFTE code, that is, $S_{n,virt}$ and $S_{n,inc,virt}$ (Subsection 4.4.3). In the MUFTE function `ConstRelDerivativesTrans` the derivatives of the constitutive relationships are computed. As in the transition condition according to equation (19), three cases are distinguished here. The derivative of the capillary pressure is approximated linearly as this was always fairly close to the analytical derivative in practise. If the virtual saturation is zero and the entry pressure has not yet been reached ($S_{n,virt} \neq S_n$ and $p_{c,min} < p_{entry}$) the derivatives of the constitutive relationships are set to zero (no non-wetting phase has entered into the material with the higher entry pressure yet). If the entry pressure is reached and we are at the interface ($S_{n,virt} \neq S_n$ and $p_{c,min} >= p_{entry}$) the relative permeability – saturation relationships are calculated using the analytical derivatives of the BROOKS-COREY relationships, see Appendix B. If the virtual saturation is equal to the non-virtual saturation (we are not at the interface or we are at the interface and consider an element which belongs to the rock type with the lower entry pressure) the mobility derivatives for S_n are calculated using a linear approximation of the $k_{r\alpha} - S_w$ relationships, see also Subsection 4.4.5. The last step is to actually compute the entries of the Jacobian which is done in the routine `ExactJacobianTrans` (Subsection 4.4.4). Here it is important to pay attention to compute the state derivatives for the accumulation terms $\frac{\partial f_\alpha}{\partial S_n}$ as discussed in Section 4.4.

$$
\begin{aligned}
\text{if } (1 - S_n < \delta) \qquad & 1 - S_n = \delta \\
\text{if } (1 - S_n > 1.0) \qquad & 1 - S_n = 1.0 \\
p_{c,min} &= f(1 - S_n) \\
S_{n,inc} &= S_n + \epsilon \\
p_{c,min,inc} &= f(1 - S_{n,inc})
\end{aligned}
$$

$$S_{n,virt} = \begin{cases} S_n & \text{if } p_{c,min} = p_c \\ 0 & \text{if } p_{c,min} < p_{entry} \\ InvPc^{II}(Pc^I(1 - S_n)) & \text{if } p_{c,min} > p_c \end{cases}$$

$$S_{n,inc,virt} = \begin{cases} S_{n,inc} & \text{if } p_{c,min,inc} = p_c \\ 0 & \text{if } p_{c,min,inc} < p_{entry} \\ InvPc^{II}(Pc^I(1 - S_{n,inc})) & \text{if } p_{c,min,inc} > p_c \end{cases}$$

$$\frac{\partial p_c}{\partial S_n} = \frac{p_c(1 - S_{n,inc,virt}) - p_c(1 - S_{n,virt})}{S_{n,inc} - S_n}$$

if $(S_{n,virt} \neq S_n)$

{

if $(p_{c,min} < p_{entry})$

$$\frac{\partial \lambda_w}{\partial S_n} = \frac{\partial \lambda_n}{\partial S_n} = \frac{\partial p_c}{\partial S_n} = 0.0$$

else

Calculate $\frac{\partial \lambda_w}{\partial S_n}$ and $\frac{\partial \lambda_n}{\partial S_n}$ analytically.

}

else

{

$$\frac{\partial \lambda_w}{\partial S_n} = \frac{1}{\mu_w} \cdot \frac{k_{rw}(1 - S_{n,inc,virt}) - k_{rw}(1 - S_{n,virt})}{S_{n,inc} - S_n}$$

$$\frac{\partial \lambda_n}{\partial S_n} = \frac{1}{\mu_n} \cdot \frac{k_{rn}(1 - S_{n,inc,virt}) - k_{rn}(1 - S_{n,virt})}{S_{n,inc} - S_n}$$

}.

4.4.2 PreCapillaryPressure

This function computes the minimum capillary pressure for a node v_i^k depending on its wetting phase saturation $S_{w,i}$ and the capillary pressure functions of the elements it belongs to

$$p_{c,min}^i = \min_{k \in E(i)} p_c(x^k, S_{w,i}),$$

where E(i) is the set of indices of the elements which have v_i as a corner, and x^k is the barycenter of element e_k.

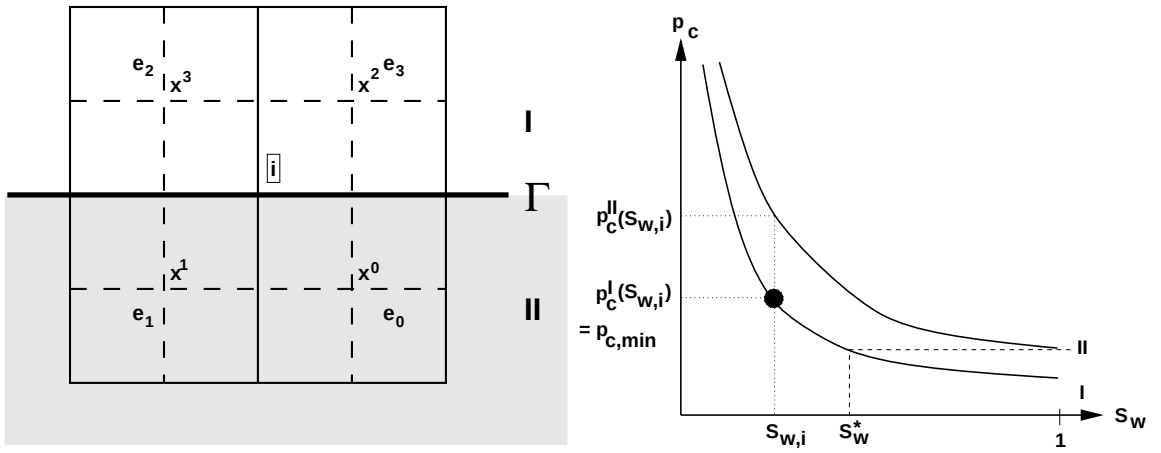

Figure 13: Evaluation of the minimum capillary pressure for node i

The principal way of computing the minimum capillary pressure for a vertex i is illustrated in Figure 13 on the left: we evaluate the capillary pressure for elements e_0 through e_3 which are adjoined to v_i with the saturation at v_i and take the minimum. As the barycenters x^0 and x^1 lie within material II, the $p_c - S_w$ function II is taken, for x^2 and x^3 it is the capillary pressure - saturation function of material I, respectively. As can be seen from Figure 13 on the right, in case of an interface the minimum capillary pressure always corresponds to the capillary pressure of the material with the lower entry pressure (the more highly permeable material).
In the code, within an outer loop over all elements of the grid, an inner loop over the vertices of each element is called. First the wetting phase saturation is limited to the interval $[\delta, 1]$, as described in the previous section. In this way, not only non-physical saturations are avoided, but also too steep gradients (e.g. for the capillary pressure - saturation function for small S_w). In the next step, the element barycenter is calculated in order to use the right $p_c - S_w$ relationship. Then the capillary pressure is calculated and compared to the minimum capillary pressure at this node so far. The evaluation is carried out elementwise, so that each node is evaluated more than once depending on the number of adjacent elements. Each time, the capillary pressure is compared to $p_{c,min}$ for this vertex and the lower value of the capillary pressure is stored in $p_{c,min}$ for the respective node [2].
First, the minimum capillary pressure is computed for the non-virtual saturation S_n. Then a value of $p_{c,min}$ for the incremented non-virtual saturation $S_{n,inc}$ is needed which will be denoted $p_{c,min,inc}$ in the following. To compute $S_{n,inc,virt}$ the interface condition is evaluated using $p_{c,min,inc}$.

for (e = FirstElement; e = LastElement; e++)
{
    for (i = 0; i < NumberOfVerticesOfElement; i++)
    {
        CalculationOfElementBarycenter;
        PropertyOfElement;
        if $(1 - S_n < \delta)$ $\quad 1 - S_n = \delta$;
        if $(1 - S_n > 1.0)$ $\quad 1 - S_n = 1.0$;
        $IN[S_w] = 1 - S_n$;
        CapillaryPressure$(IN[S_w])$;
        if $(OUT[p_c] < p_{c,min})$ $p_{c,min} = OUT[p_c]$;
        $S_n = S_n + \epsilon$
        if $(1 - S_n < \delta)$ $\quad 1 - S_n = \delta$;
        if $(1 - S_n > 1.0)$ $\quad 1 - S_n = 1.0$;
        $IN[S_w] = 1 - S_n$;
        CapillaryPressure$(IN[S_w])$;
        if $(OUT[p_c] < p_{c,min,inc})$ $p_{c,min,inc} = OUT[p_c]$;
    }
}

4.4.3 TransitionCondition

This abbreviation is used for the implementation of the transition condition. The formerly computed minimum capillary pressure is used to evaluate the saturation $S_{n,i,k}$ for vertex v_i with respect to element e_k. According to the transition condition (19) the saturation $S_{n,i,k}$ can be computed as follows:

$$S_{n,i,k} = \begin{cases} S_{n,i} & \text{if } p_c(x^k, 1 - S_{n,i}) = p^i_{c,min} \\ 0 & \text{if } p^i_{c,min} < p_c(x^k, 1) \\ 1 - S_w & \text{where } S_w \text{ solves } p_c(x^k, S_w) = p^i_{c,min} \end{cases}$$

The effect of this condition for a vertex v_i at the interface with respect to an element which belongs to material II (higher entry pressure) is illustrated in Figure 14.

for (e = FirstElement; e = LastElement; e++)
{
    PropertyOfElement;
    $IN[S_w] = 1.0$;
    CapillaryPressure$(IN[S_w])$;
    $p_{entry} = OUT[p_c]$;

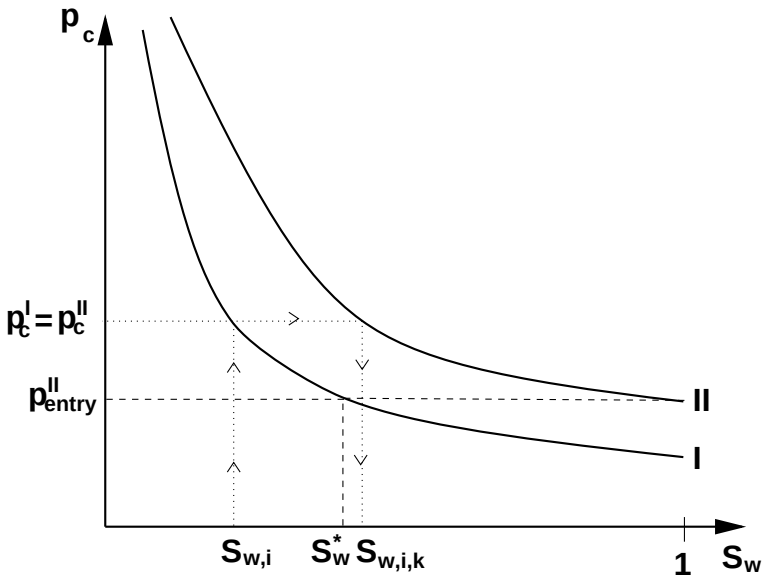

Figure 14: Transition condition at the interface

```
for (i = 0; i < NumberOfSubcontrolVolumes; i++)
{
        IN[S_w] = 1 − S_n;
        CapillaryPressure;
        if (OUT[p_c] = p_c,min)
        {
                S_n,virt = S_n;
                p_c = p_c,min;
        }
        else
        {
                if (p_c,min < p_entry)
                {
                        S_n,virt = 0.0;
                        p_c = p_c,min;
                }
                else
                {
                        InvertPC;
                }
```

```
                }
        }
}
```

The interface condition is evaluated not only for the stored saturation S_n but also for the incremented saturation $S_{n,inc}$ with respect to the minimum capillary pressure for $S_{n,inc}$, i.e. $p_{c,min,inc}$. This is also done within an outer loop over all the elements including an inner loop over the subcontrol volumes of this element. For simplicity's sake, the description of these loops is omitted in the following.

```
        {
                IN[S_w] = 1 - S_n,inc
                CapillaryPressure(IN[S_w]);
                if (OUT[p_c] = p_c,min,inc)
                {
                        S_n,inc,virt = S_n,inc;
                        p_c = p_c,min,inc;
                }
                else
                {
                        if (p_c,min,inc < p_entry)
                        {
                                S_n,inc,virt = 0.0;
                                p_c = p_c,min,inc;
                        }
                        else
                        {
                                InvertPC;
                        }
                }
        }
```

As can be seen from Figure 15, the constitutive relationships are evaluated with two (or more) different saturations for the interfacial nodes depending on the properties of the respective elements.

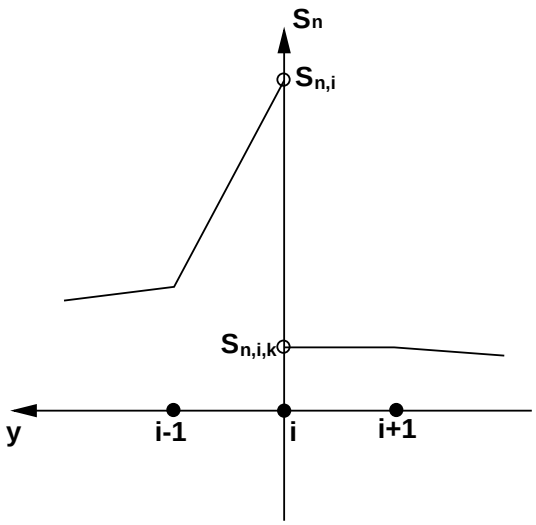

Figure 15: Saturations at the interface

4.4.4 ExactJacobianTrans

Here the element Jacobian is computed almost analytically or even analytically according to the equations in Appendices C.1, B and A. Therefore the derivatives calculated in `ConstRelDerivativesTrans` are needed. First the accumulation term as well as the sink/source term is added according to equation (42). As described there, the derivative of the accumulation term $\frac{\partial f_\alpha}{\partial S_n}$ is set to zero if the entry pressure of the material is not yet reached.
In the last step the internal flux term is computed according to Appendix C.1 and added to the actual matrix entry. As this part represents just the coding of the equations (39) no further explanations are given here.

4.4.5 ConstRelDerivativesTrans

This function computes the derivatives of

- the capillary pressure ($\frac{\partial p_c}{\partial S_w}$)
- the densities ($\frac{\partial \rho_w}{p_w}$, $\frac{\partial \rho_n}{\partial p_w}$, $\frac{\partial \rho_n}{\partial S_n}$)
- and the mobilities ($\frac{\partial \lambda_w}{\partial p_w}$, $\frac{\partial \lambda_n}{\partial p_w}$, $\frac{\partial \lambda_w}{\partial S_n}$, $\frac{\partial \lambda_n}{\partial S_n}$).

In our case, both phases are incompressible ($\frac{\partial \rho_\alpha}{\partial p_w} = \frac{\partial \rho_\alpha}{\partial S_n} = 0$) and the mobilities are not dependent on the pressure ($\frac{\partial \lambda_\alpha}{\partial p_w} = 0$). For this reason, only the derivatives $\frac{\partial p_c}{\partial S_n}$, $\frac{\partial \lambda_w}{\partial S_n}$, and $\frac{\partial \lambda_n}{\partial S_n}$ have to be considered.
As explained in Chapter 4.4.1, these derivatives have to be set to zero if the virtual saturation is not equal to the non-virtual saturation and the entry pressure has not yet been reached at the interface, that is $S_{n,virt} \neq S_n$ and $p_{c,min} < p_{entry}$. If the entry pressure of the less permeable material is reached at the interface, the mobility derivatives of the elements belonging to the material with the higher entry pressure are calculated with the help of the analytical derivatives of the relative permeability - saturation relationships according to BROOKS and COREY as derived in appendix B. If on the other hand we are at a node within one rock type or at the interface considering an element which belongs to the material with the lower entry pressure, the mobility derivatives are linearly approximated.

dpcdSn = $(p_c(1 - S_{n,inc,virt}) - p_c(1 - S_{n,virt}))/(S_{n,inc} - S_n)$
if $(S_{n,virt} \neq S_n)$
{
if $(p_{c,min} < p_{entry})$
dlambdawdSn=dlambdandSn=dpcdSn = 0.0
else
Calculate dlambdawdSn and dlambdandSn with help of the analytical derivatives of krw and krn.
}
else
{
dlambdawdSn = $1/\mu_w * (k_{rw}(1 - S_{n,inc,virt}) - k_{rw}(1 - S_{n,virt}))/(S_{n,inc} - S_n)$
dlambdawdSn = $1/\mu_n * (k_{rn}(1 - S_{n,inc,virt}) - k_{rn}(1 - S_{n,virt}))/(S_{n,inc} - S_n)$
}.

4.4.6 Hierarchy of the Implemented Functions

The transition condition is implemented not only in the term of the right hand side (defect term), but also in the matrix of the equation system. For the matrix, the exact linearization has to be taken into account, too.

The defect term is composed in the function TAssembleDefect. First, the minimum capillary pressures are needed in order to evaluate the transition condition. Then, the static data (like gravity, p_w, p_n, source term) are evaluated at the nodes, the absolute permeability is calculated at the barycenter. In EvaluateStaticDataAtIntegrationPoints, the values on the boundary faces are evaluated. UpdateIntegrationPoints computes the mobilities at the integration points and the gradients of the extended Darcy velocities. For the other functions, the name indicates the purpose of the function.

TAssembleDefect
{
PreCapillaryPressure;
TransitionCondition
EvaluateStaticDataAtNodes;
EvaluateStaticDataAtIntegrationPoints;
UpdateIntegrationPoints;
EvaluateUpwindSwitches;
ComputeStorageTerm;
ComputeFluxTerm;
ComputeSourceTerm;
AccumulateToGlobalVector;
PostCapillaryPressures;
}

The transition condition with exact linearization of the PDE's is implemented in the function TAssembleMatrixExact in the MUFTE code. Here, the matrix of the equation system is assembled. First, the minimum capillary pressures are needed in order to evaluate the transition conditions for S_n and $S_{n,inc}$. Then, for the matrix entries, the exact Jacobian is needed which itsself needs the calculation of some derivations for the primary variables computed in ConstRelDerivativesTrans. The function PostCapillaryPressures frees the memory allocated by the function PreCapillaryPressure.

TAssembleMatrixExact
{
PreCapillaryPressure;
TransitionConditionForSn
TransitionConditionForSninc

```
        ExactJacobianTrans
        {
                ConstRelDerivativesTrans
        }
        EvaluateStaticDataAtNodes;
        EvaluateStaticDataAtIntegrationPoints;
        UpdateIntegrationPoints;
        AccumulationToGlobalMatrix;
        EliminationOfDirichletBoundaryConditionsForPrimaryVariables;
        PostCapillaryPressures;
}
```

4.5 The NEWTON-RAPHSON Method

To solve the highly nonlinear system of partial differential equations

$$\mathbf{F}(\mathbf{x}) = \mathbf{0}$$

the damped inexact NEWTON-RAPHSON method is used [16].
The algorithm used is the following:

Choose $\mathbf{x^{k+1,0}}$; set $r = 0$;
while $(\frac{(\|\mathbf{F}(\mathbf{x^{k+1,r}})\|_2}{\|\mathbf{F}(\mathbf{x^{k+1,0}})\|_2} > \epsilon_{nl})$
{
Solve $\mathbf{K}(\mathbf{x^{k+1,r}})\mathbf{u} = -\mathbf{F}(\mathbf{x^{k+1,r}})$ with accuracy ϵ_{lin};
$\mathbf{x^{k+1,r+1}} = \mathbf{x^{k+1,r}} + \eta\mathbf{u}$;
$r = r + 1$;
}.

Here, $\mathbf{F}(\mathbf{x^{k+1,r}})$ represents the defect term at time step $k+1$ and iteration r. The constant ϵ_{nl} is the accuracy criterion for the nonlinear solution, ϵ_{lin} that for the linear solution. The aim is to choose a damping factor $\eta = \left(\frac{1}{2}\right)^q$ such that

$$\| \mathbf{F}(\mathbf{x^{k+1,r+1}}) \|_\mathbf{2} \leq [\mathbf{1} - \beta \cdot \eta] \| \mathbf{F}(\mathbf{x^{k+1,r}}) \|_\mathbf{2}$$

is fulfilled for the smallest possible $q \in \{0, 1, \ldots, 6\}$. The factor β is usually set to $\frac{1}{4}$.
For time discretization a fully implicit Eulerian approach is used. Thus, the storage term is discretized as

$$\frac{\partial S}{\partial t} \approx \frac{S^{k+1} - S^k}{\Delta t^{k+1}}$$

The time step is halved if no q is found within 6 line searches, and it is doubled if this q is found within the first line search. Within the code, a starting value for the time step as well as a minimum and a maximum time step are fixed. Thus, the size of the actual time step can be interpreted as an indicator for the convergence behaviour of the nonlinear algorithm.
As linear solver, a BiCGStab solver with V multigrid cycle as preconditioner is used.

5 Examples

To show the correctness and the efficiency of the newly developped code using the BOX method with interface condition and consistent linearization of the PDE's, numerical simulations of some examples using MUFTE-ug are carried out. First, the method is applied to a simple 1D example using principle (not necessarily physical) sets of parameters and constitutive relationships to investigate the behaviour of the method under extreme conditions in order to find the sensitive parameters. In a second step, the arbitrarily chosen parameters and relationships are replaced by realistic ones using the same simple 1D example. Comparisons of the results as well as performance studies are carried out. Finally, the code is applied to a 2D example corresponding to an experiment actually performed in the VEGAS research facility of the University of Stuttgart.

5.1 1D Examples

In this section, the BOX method with interface condition and consistent linearization of the PDE's is applied to a simple 1D example. In Section 5.1.1 some theoretical sets of constitutive relationships are used whereas in Section 5.1.2 realistic parameters from a laboratory experiment are taken.
The setup and boundary conditions of the example are given in Figure 16. A vertical column which is initially fully water saturated and under a constant pressure (initial values $S_n = 0.0$, $p_w = 2.0 \cdot 10^5$ Pa) is infiltrated by DNAPL from the top boundary. The numerical description of the example is done in 2D, but as the parameters are constant along the x coordinate it represents in fact a 1D example. The column consists of two different materials, denoted by I and II. Material I has a vertical length of 0.75 m, whereas material II has a length of 0.5 m. The width of the column is 0.5 m. On the sides of the column, we impose noflow boundaries (NEUMANN type boundaries). The bottom boundary is characterized by a constant pressure (which is equal to the initial pressure) and by full water saturation ($S_n = 0.0$, DIRICHLET boundary). At the top, DNAPL (TCE, density 1621 $\frac{kg}{m^3}$, viscosity $9 \cdot 10^{-4} m^2/s$) infiltrates at a rate of $S_n = -0.05$, the water pressure is equal to zero. For water, a density of 998 $\frac{kg}{m^3}$ and a viscosity of $10^{-3} \frac{m^2}{s}$ are taken into account.

5.1.1 Principle parameter studies

To investigate the influence of differerent constitutive relationships, which turned out to be the crucial point for the performance of the trans method with numerical linearization of the PDE's, some principle parameter studies are carried out. For all cases,

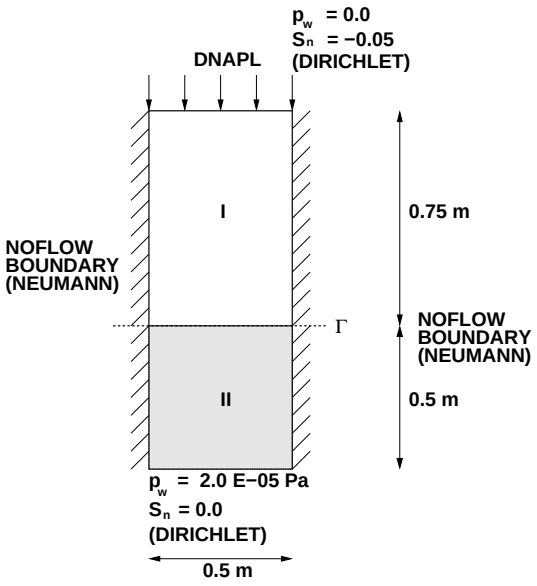

Figure 16: Setup and boundary conditions of the 1D example

the porosity ϕ of both materials is equal to 0.4, the absolute permeability K of material I is $5 \cdot 10^{-10} m^2$ and that of material II $3 \cdot 10^{-11} m^2$.
As for the numerical trans method the slope of the k_{rw}-S_w function at $S_w = 1$ had turned out to be the critical point [15], polynomial functions for the relative permeability - saturation relationships are taken which make this slope easily controllable (it is simply equal to the exponent e). For the capillary pressure, the BROOKS-COREY relationships are taken, see Section 3.1.
To sum up, we are dealing with the following functions:

$$p_c^I(S_e) = p_d^I \cdot S_e^{\frac{2+3\lambda^I}{\lambda^I}} \tag{43}$$

$$p_c^{II}(S_e) = p_d^{II} \cdot S_e^{\frac{2+3\lambda^{II}}{\lambda^{II}}} \tag{44}$$

$$k_{rw}(S_e) = S_e^e \tag{45}$$

$$k_{rn}(S_e) = (1 - S_e)^e \tag{46}$$

The parameters λ^I and λ^{II}, the difference of the entry pressures $\Delta p_d = p_d^I - p_d^{II}$ and the exponent e will be varied during the case studies. In the end, a "worst case" comparison is made combining the parameters with the worst performance in the single parameter experiments. The parameters for the comparisonal studies can be found in Table 1.

variation of	$p_d[Pa]$			$\lambda[-]$			$e[-]$			"worst case"
	a	b	c	a	b	c	a	b	c	
$p_d^I[Pa]$	100	100	100	100	100	100	100	100	100	100
$p_d^{II}[Pa]$	200	500	800	200	200	200	200	200	200	800
$\lambda^I = \lambda^{II}[-]$	0.4	0.4	0.4	0.4	1.5	3.0	0.4	0.4	0.4	3.0
$e[-]$	0.25	0.25	0.25	0.25	0.25	0.25	0.25	1.0	4.0	4.0

Table 1: Parameters used for the principle parameter studies

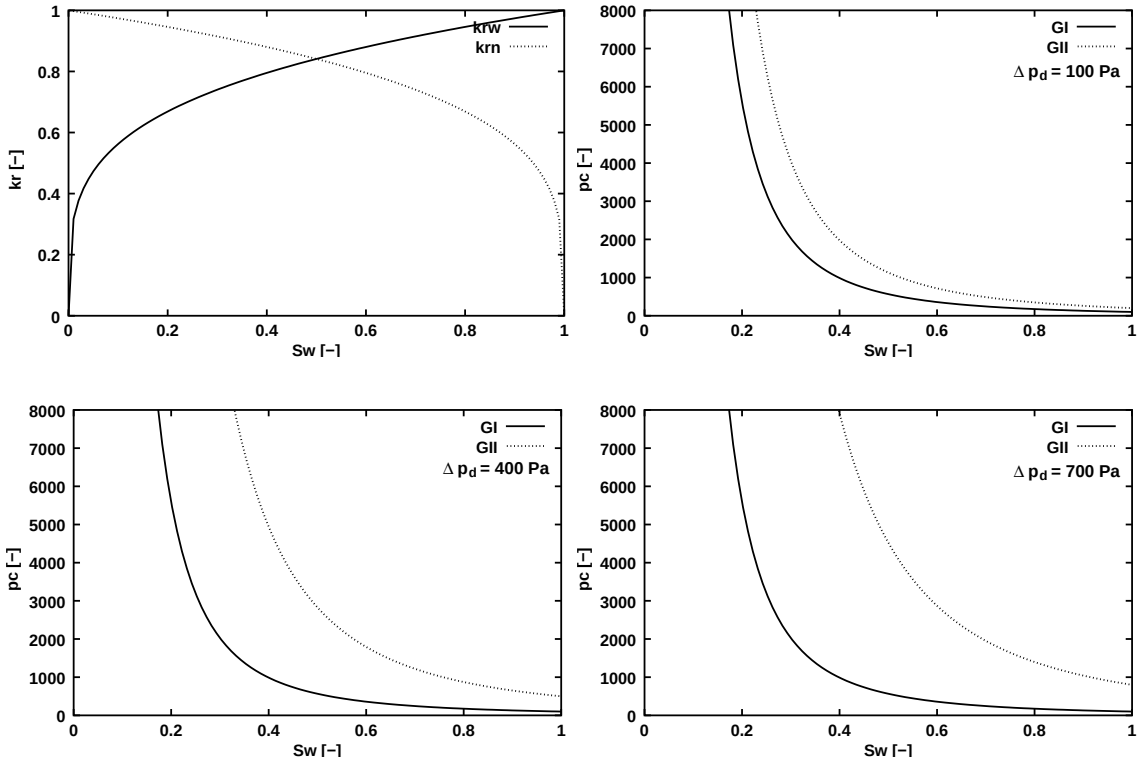

Figure 17: The $k_r - S_w$ and $p_c - S_w$ relationships for different values of the difference in entry pressures (case a: top right, b: bottom left, c: bottom right)

The curves of the respective constitutive relationships can be taken from Figures 17 through 20.
The performances of these parameter studies are displayed and discussed in Subsection 5.1.3. As we are dealing with principle parameter studies where just the differences in the performances are important, a fairly coarse mesh and a large time step are cho-

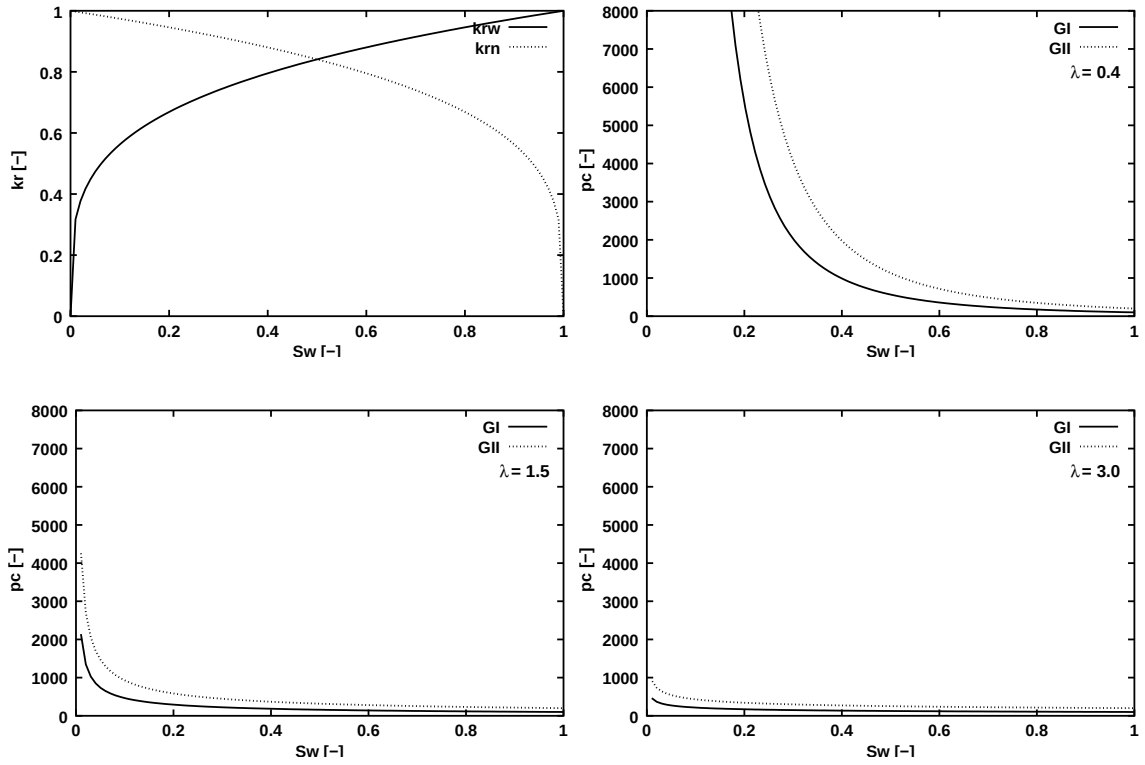

Figure 18: The $k_r - S_w$ and $p_c - S_w$ relationships for different values of the BROOKS-COREY parameter λ (case a: top right, b: bottom left, c: bottom right)

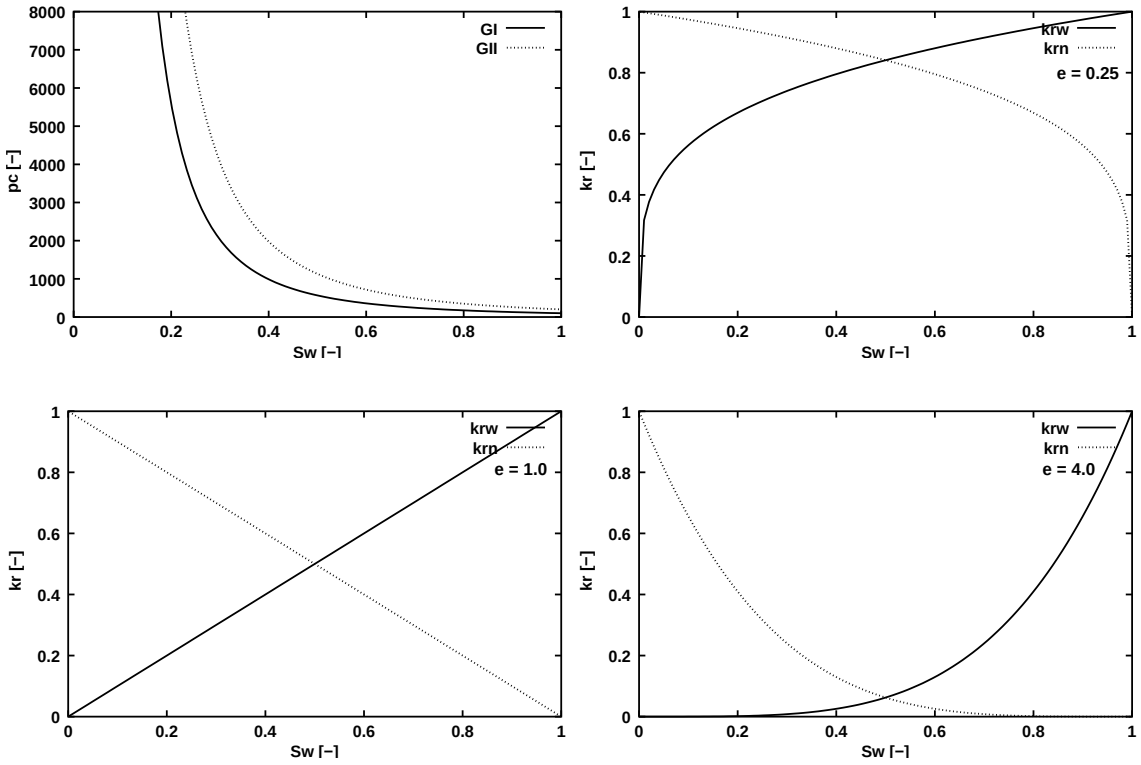

Figure 19: The $k_r - S_w$ and $p_c - S_w$ relationships for different values of the exponent e of the polynomial k_r functions (case a: top right, b: bottom left, c: bottom right)

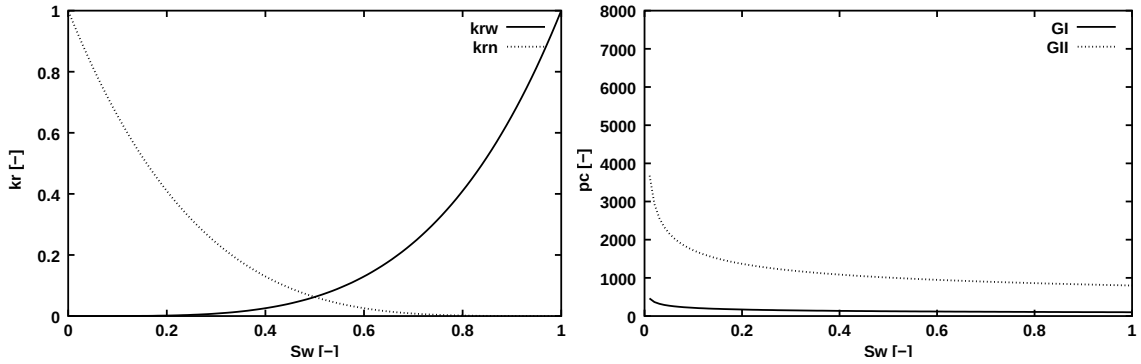

Figure 20: The $k_r - S_w$ and $p_c - S_w$ relationships for the "worst case" combination of parameters, $\Delta p_d = 700$ Pa, $\lambda = 3.0$, $e = 4.0$.

	$\phi[-]$	$K[m^2]$	$\lambda[-]$	$p_d[Pa]$	S_{wr}
I	0.39	$4.6 \cdot 10^{-10}$	2.0	200	0.05
II	0.35	$3.1 \cdot 10^{-11}$	2.3	700	0.015

Table 2: Parameter values for coarse and medium sand taken from a VEGAS experiment

sen. The elements have a size of 0.25 x 0.25 m^2, the initial and maximum time step are set to 100 s. If the time step gets smaller than 0.1 s (by halving the time step if the maximum number of line searches is exceeded, see Section 4.5), the computation is aborted.

5.1.2 Example using experimental data

For the same setup, boundary and initial values as for the principle parameter studies, see Figure 16, experimental parameter sets and constitutive relationships are taken to get a feeling for the performance of the transition condition with exact linearization for practical applications. The parameter values are taken from an experiment carried out at the VEGAS research facility at the University of Stuttgart [3]. The values are given in Table 2.
The time step was chosen according to the COURANT-FRIEDRICHS-LEWY criterion (chosen by determining the front velocity in preliminary investigations). Within one time step, the fluid should progress by no more than one element. With this restriction, a time step of 40 s and a mesh width of 20.83 cm was chosen.

5.1.3 Results of the 1D investigations

Principle parameter studies The results of experiments with the eight parameter combinations of Table 1 for the principle parameter studies are shown in Figures 21 through 28. The performances of the BOX method without transition condition and with numerical linearization("standard"), the BOX method with interface condition and numerical linearization ("trans numeric") and the BOX method with interface condition and exact linearization of the PDE's ("trans exact") are compared.
In the first picture (upper left) of a figure, the simulation time is plotted as a function of the computing time. The larger the simulation time for a certain computing time, the better is the performance of the algorithm. The other three pictures show the number of nonlinear iterations per time step of the newton solver (see Section 4.5) over the simulation time in the following order: upper right picture: standard, bottom left: trans numeric and bottom right: trans exact.

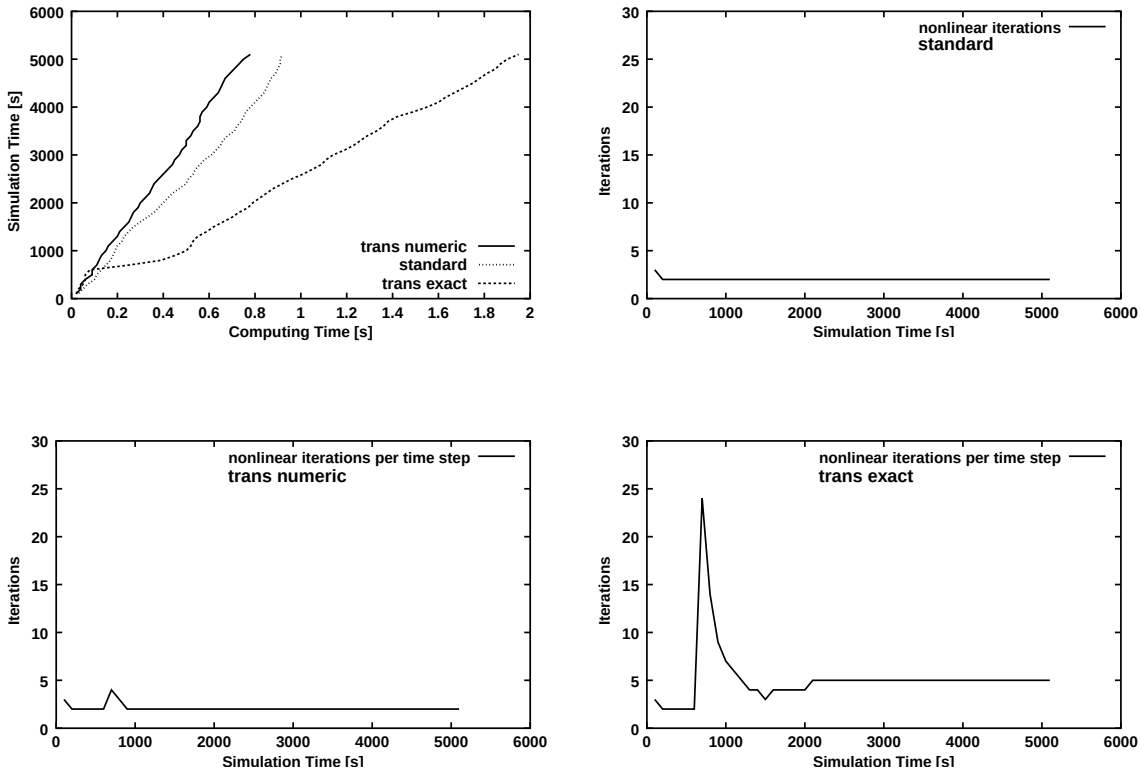

Figure 21: Simulation time as a function of computing time and number of nonlinear iterations per time step for $\Delta p_d = 100$ Pa, $\lambda = 0.4$, and $e = 0.25$

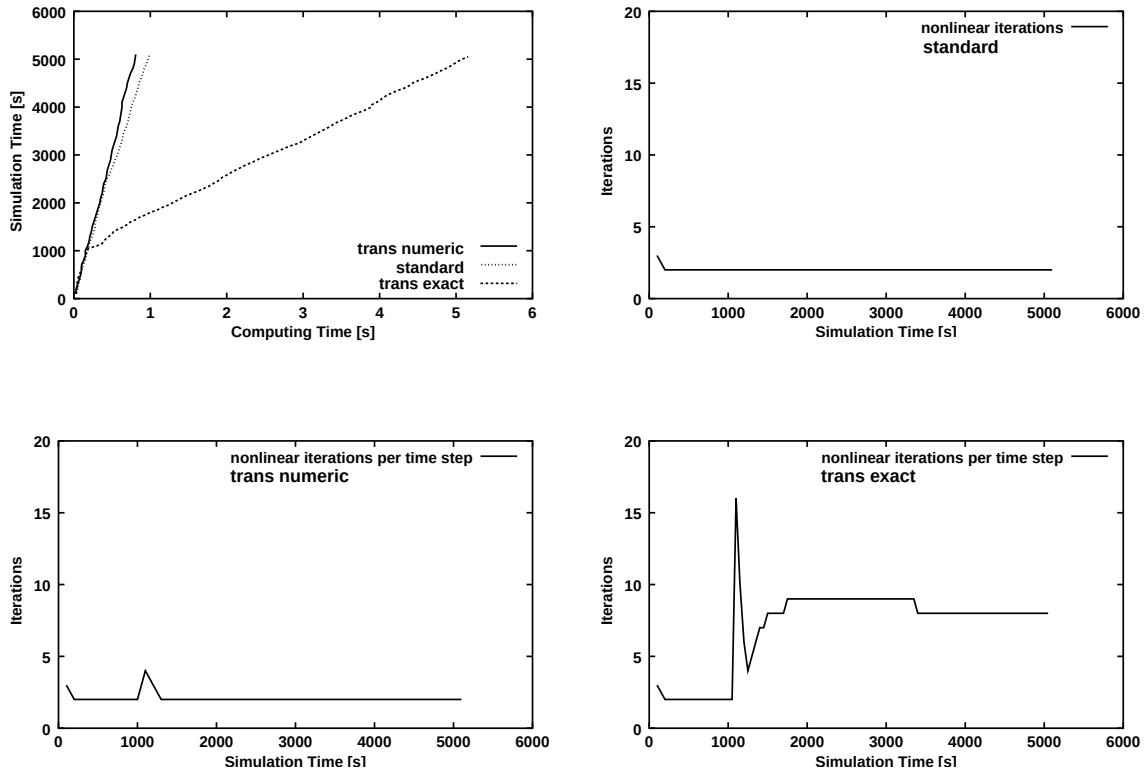

Figure 22: Simulation time as a function computing time and number of nonlinear iterations per time step for $\Delta p_d = 400$ Pa, $\lambda = 0.4$, and $e = 0.25$

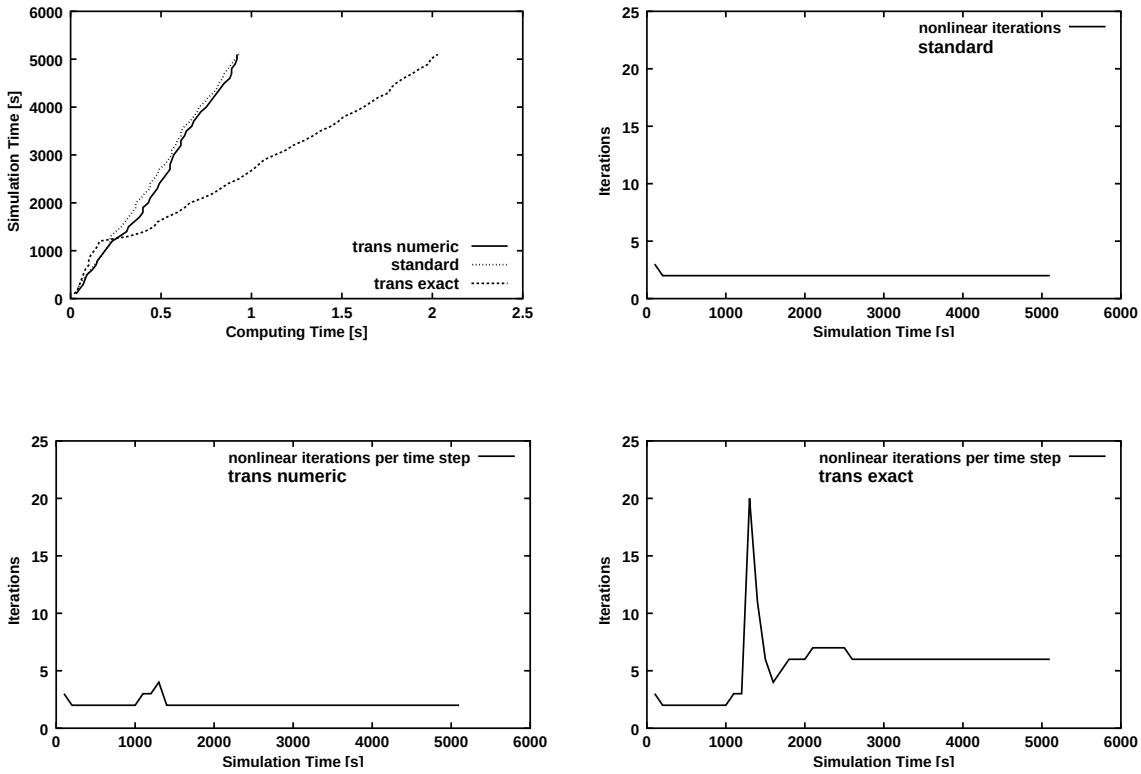

Figure 23: Simulation time as a function of computing time and number of nonlinear iterations per time step for $\Delta p_d = 700$ Pa, $\lambda = 0.4$, and $e = 0.25$

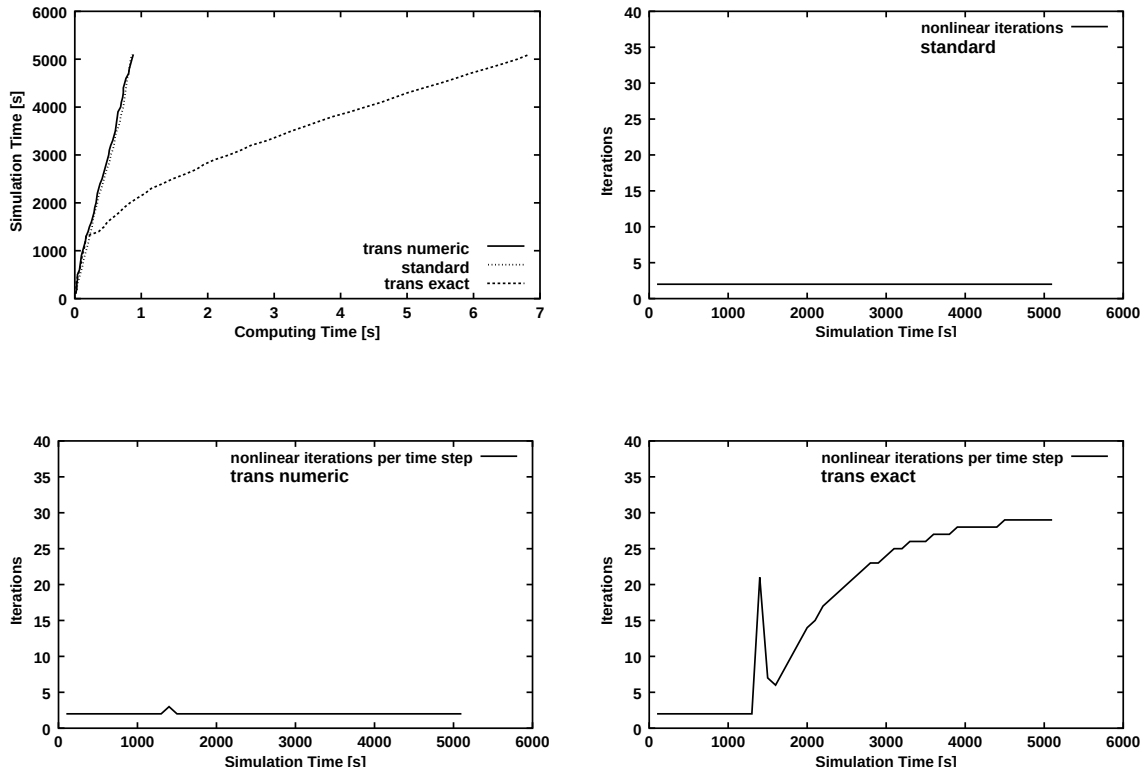

Figure 24: Simulation time as a function of computing time and number of nonlinear iterations per time step for $\Delta p_d = 100$ Pa, $\lambda = 1.5$, and $e = 0.25$

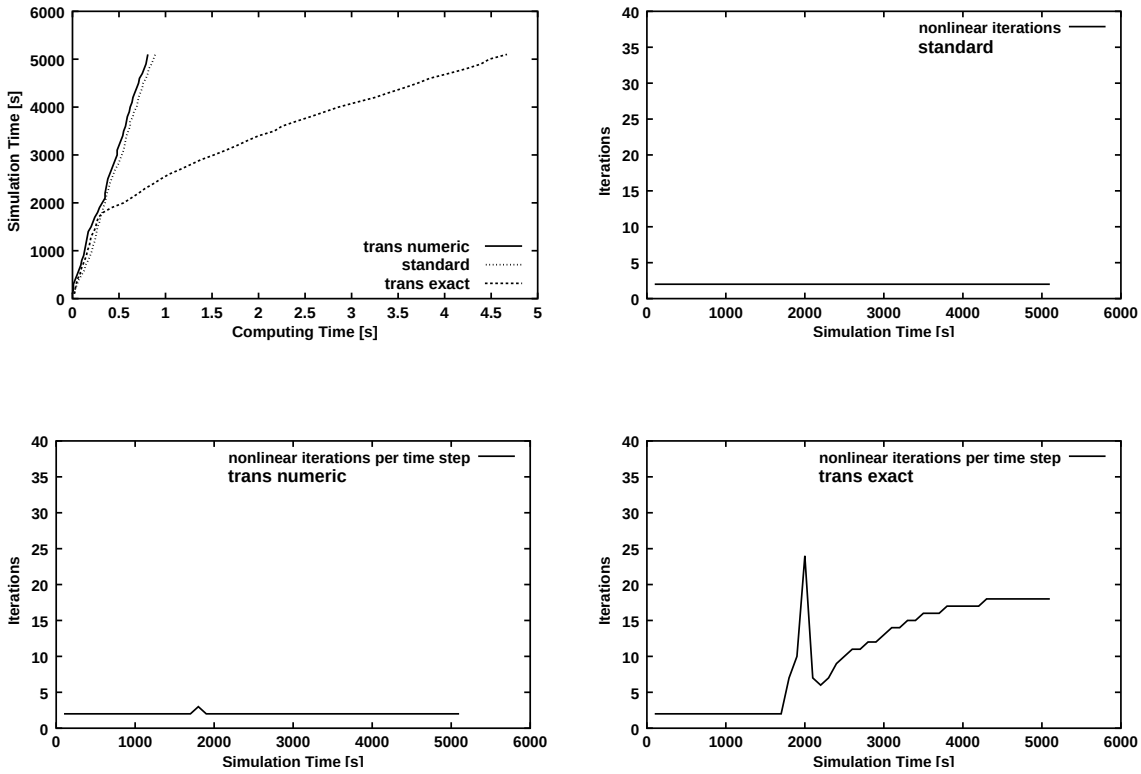

Figure 25: Simulation time as a function of computing time and number of nonlinear iterations per time step for $\Delta p_d = 100$ Pa, $\lambda = 3.0$, and $e = 0.25$

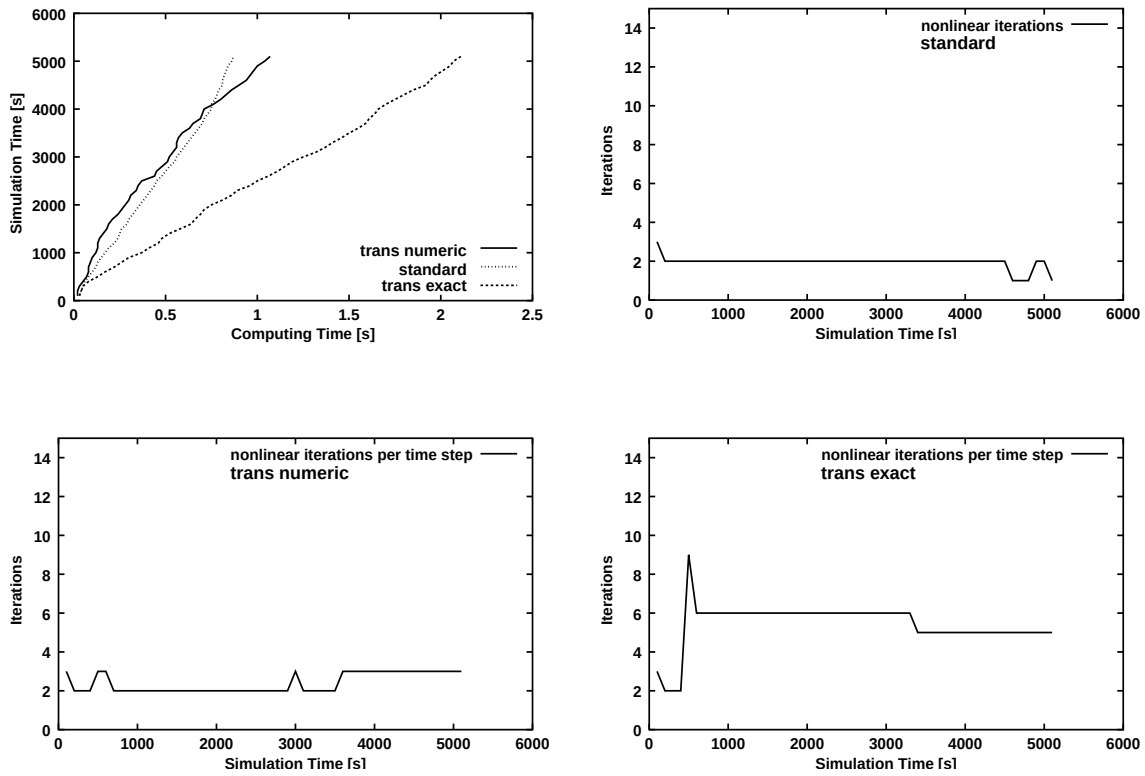

Figure 26: Simulation time as a function of computing time and number of nonlinear iterations per time step for $\Delta p_d = 100$ Pa, $\lambda = 0.4$, and $e = 1.0$

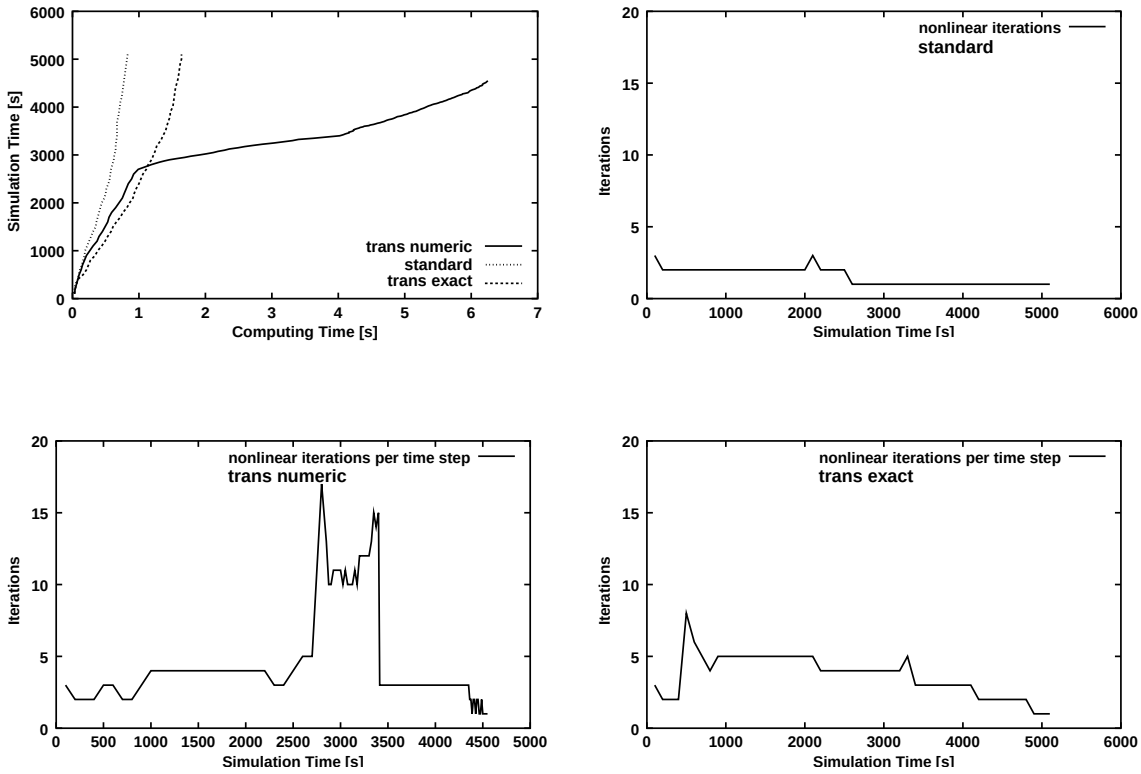

Figure 27: Simulation time as a function of computing time and number of nonlinear iterations per time step for $\Delta p_d = 100$ Pa, $\lambda = 0.4$, and $e = 4.0$

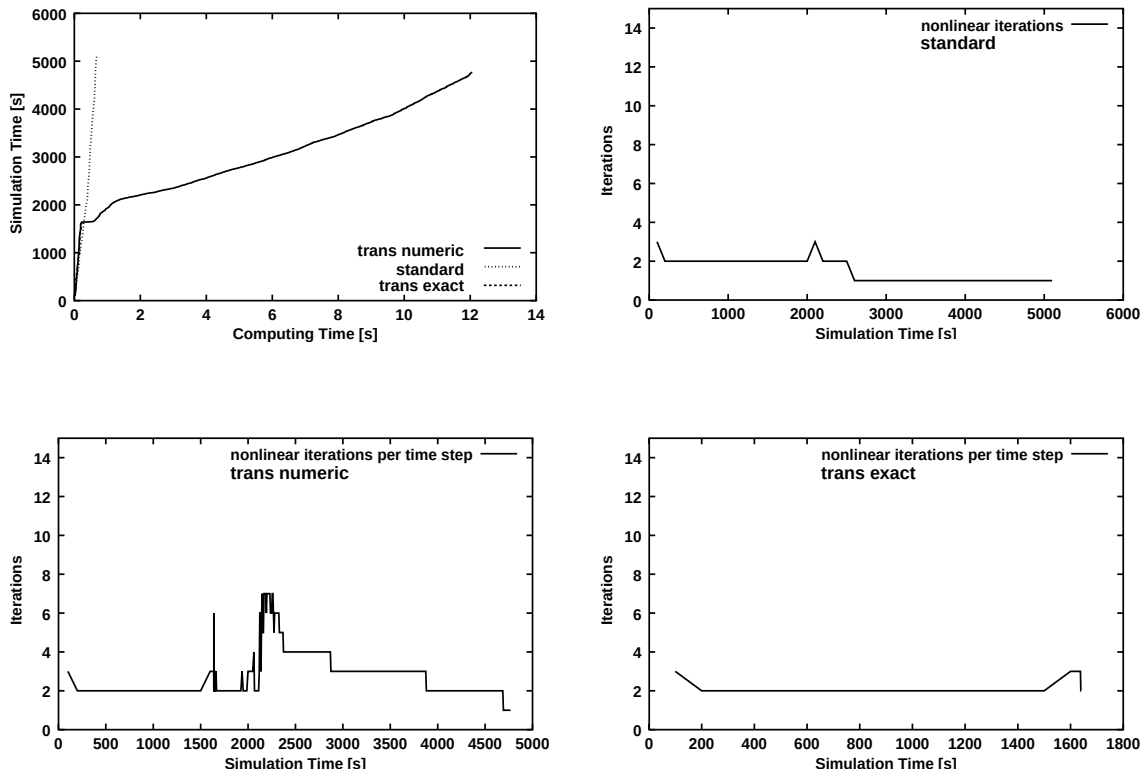

Figure 28: Simulation time as a function of computing time and number of nonlinear iterations per time step for $\Delta p_d = 700$ Pa, $\lambda = 3.0$, and $e = 4.0$

How can we explain this behaviour? From Figures 21 through 28 it gets obvious that for some parameter combinations the "trans numeric" method works best. For other parameter sets, it is the "standard" method. However, if we wish to model the interface in a physically correct manner, we have to restrict ourselves to the schemes with interface condition, i.e. "trans numeric" and "trans exact". But even for these cases, it seems that neither method is clearly preferable: sometimes "trans numeric" performs better than "trans exact", sometimes it is the other way around.
As only the methods with the same type of linearization can be directly compared, the "standard" method is compared to the "trans numeric" method in a first step as they both use numerical linearization. In a second step, the general dependence of the "trans exact" method on the different parameters is investigated in order to explain its behaviour for the different sets of parameters.
At first, we might expect that in general "trans numeric" should perform better than "standard", because with the trans numeric scheme

- there are fewer oscillations of the upwinding node and
- one can detect smaller gradients of the capillary pressure at the interface, see e.g. Figure 41.

However, the results show, that this is not always the case. By increasing Δp_d, the difference between the entry pressures of the two materials (Figures 21 through 23), one can see that the performance of "trans numeric" becomes worse in comparison with to "standard". The number of nonlinear iterations increases when reaching the interface but normalizes again thereafter. This is due to the higher value of the variation δ' of the virtual saturation $S_{n,virt}$ for higher ratios of the entry pressures $\frac{p_d^{II}}{p_d^I}$, see Appendix D.
Changing the value of λ for the capillary pressure - saturation relationship does not have a significant influence on the performance of either the "trans numeric" or the "standard" scheme ("trans numeric" performs better than "standard"). The figures to be compared are 21, 24, and 25. For the number of nonlinear iterations per time step we make the same observations as for the variation of the entry pressure: a constant behaviour for "standard" and a slight increase in the number of nonlinear iterations per time step for "trans numeric" at the interface.
The parameter which clearly has the largest influence on the performance, is the exponent e of the relative permeability - saturation relationships. Increasing λ enormously worsens the performance of "trans numeric", see also Figure 1. As mentioned there this is due to the steep gradient of the $k_{rw} - S_w$ function at $S_w = 1$. Given in Figure 29, the variation of the primary variable S_w is generally higher for the virtual than for the nonvirtual saturation. Considering the number of nonlinear iterations per time step,

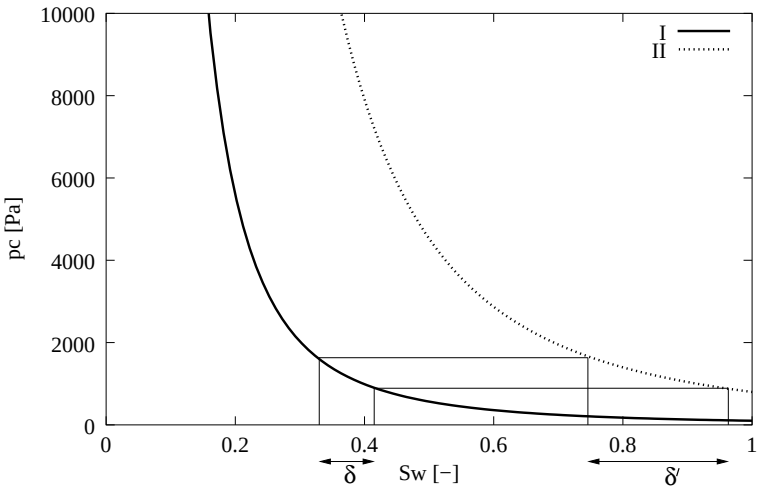

Figure 29: Variation δ of the nonvirtual and of the virtual (δ') saturation

an enourmous increase for "trans numeric" can be recognized when reaching the interface with increasing exponent (and thus increasing steepness of $k_{rw} - S_w$ at $S_w = 1$). Combining the parameters that have performed worst, "trans exact" does not even reach the end of the simulation time, as the execution is aborted due to halfening of the time step up to a value smaller than the minimum time step.
Now, the dependence of the behaviour of the "trans exact" on the parameters Δp_d, λ, and e is to be investigated. According to the equations (43) through (46) and appendix A, the steepness of the $k_{rw} - S_w$ relationships at $S_w = 1$ is equal to

$$\frac{\partial k_{rw}(S_{w,virt})}{S_w}(S_w = 1) = e \cdot \left(\frac{p_d^{II}}{p_d^I}\right)^{\lambda^{II} e} \cdot \frac{\lambda^{II}}{\lambda^I}. \tag{47}$$

For the "trans numeric" method, the corresponding gradient would be equal to the exponent e, which can be derived from (45):

$$\frac{\partial k_{rw}(S_w)}{S_w}(S_w = 1) = e \cdot S_w^{e-1}(S_w = 1) = e. \tag{48}$$

Whereas for "trans numeric" the critical steepness is only dependent on e, it is dependent on different parameters for "trans exact". The corresponding gradient becomes steep for

1. large values of e (even more so than for the "trans numeric", as e can be found in the exponent of the ratio of the entry pressures ($p_d^{II}/p_d^I > 1$)),

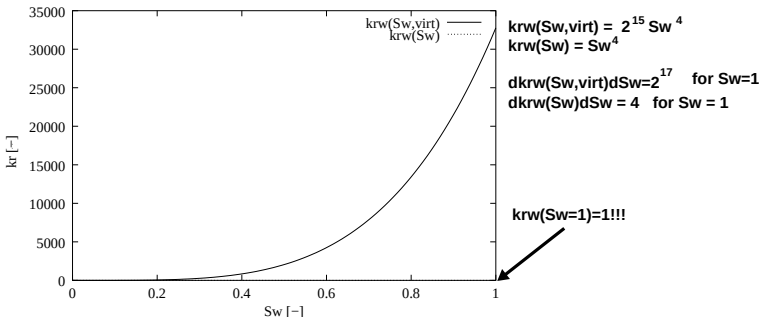

Figure 30: $k_{rw} - S_w$ functions and steepnesses at $S_w = 1$ for the "worst case" set of parameters

2. large values of the ratio $\frac{\lambda^{II}}{\lambda^I}$, especially large λ^{II}, as λ^{II} is in the exponent of $p_d^{II}/p_d^I > 1$,

3. large ratios of $\frac{p_d^{II}}{p_d^I}$.

An example is provided in Figure 30. Here the slopes at $S_w = 1$ are computed for the "worst case" parameter set.
Item 3) can be observed when comparing Figures 21 to 23. The same is valid for increasing λ values (item 2), Figures 21, 24 and 25). With increasing e the performance of "trans exact" gets better than the performance of "trans numeric". The reason can also be taken from the upper enumeration: whereas for "trans numeric" only e is decisive for the steepness of $k_{rw} - S_w$, it is the combination of all parameters for "trans exact". In other words: an unfavourable value for e (high value) can be compensated by favourable values of the other parameters (low $\frac{\lambda^{II}}{\lambda^I}$, λ^{II} and $\frac{p_d^{II}}{p_d^I}$) for "trans exact".
In special cases, it can also be the NEWTON algorithm which encounters problems and an improvement of the efficiency can be reached by the use of the Trust Region algorithm. This possibility will be investigated in Chapter 6.

The VEGAS experiment For this experiment, the performances of "standard", "trans numeric" and "trans exact" are compared (Figure 31), exactly in the same way as before for the principle parameter studies.
The difference between this experiment and the principle studies lies in the relative permeability - saturation relationship. Whereas polynomial relationships were chosen for the principle parameter studies, the BROOKS-COREY relations are applied now. The decisive difference lies in the fact that using BROOKS-COREY relations for the

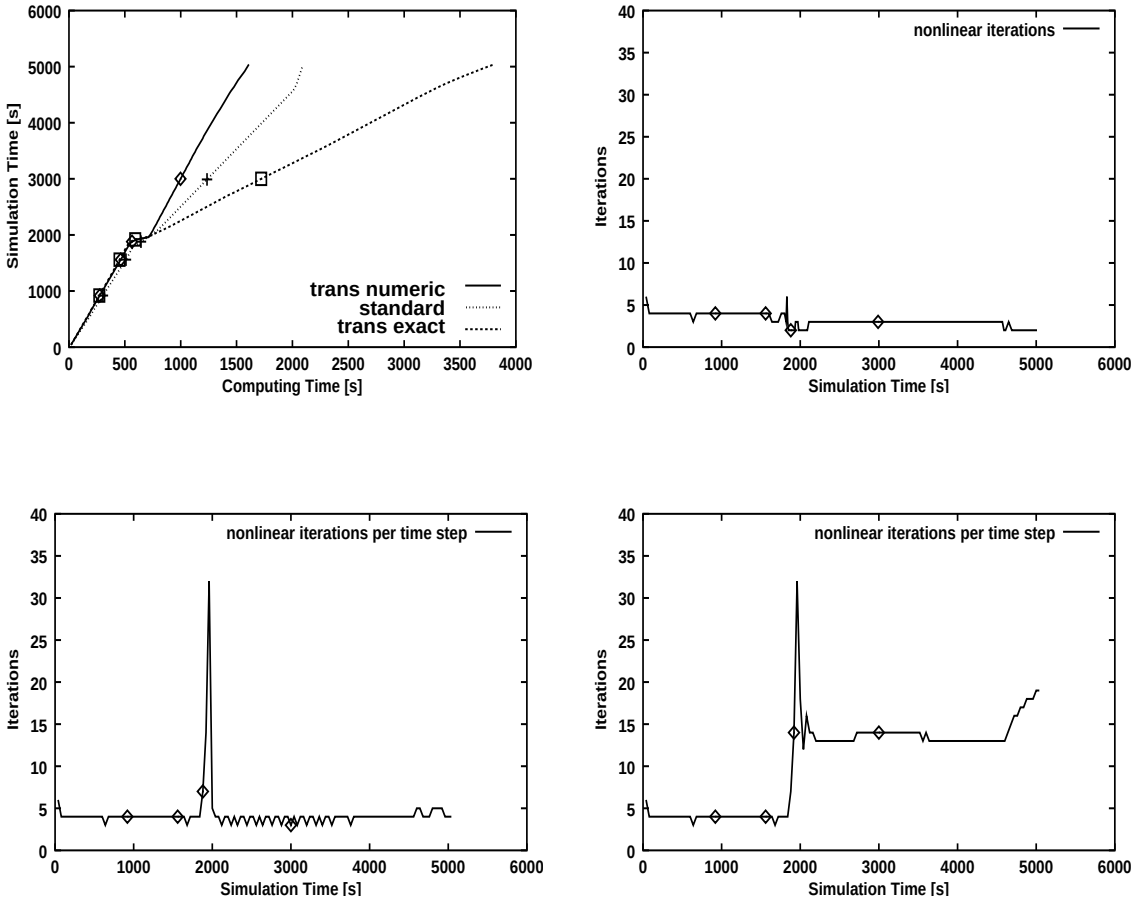

Figure 31: Simulation time over computing time and nonlinear iterations for "standard" (upper right), "trans numeric" (bottom left) and "trans exact" (bottom right)

relative permeability - saturation and capillary pressure - saturation functions the two types of constitutive relationships are no longer independent of each other, but coupled via the parameter λ. Given in Appendix B, the steepness of the $k_{rw} - S_w$ function at $S_w = 1$ is in this case equal to

$$\frac{\partial k_{rw}(S_{w,virt})}{S_w}(S_w = 1) = \frac{2 + 3\lambda^*}{\lambda^*} \left(\frac{p_d^{II}}{p_d^I} \right)^{\lambda^{II} \frac{2+3\lambda^*}{\lambda^*}} \cdot \frac{\lambda^{II}}{\lambda^I}, \tag{49}$$

where $*$ represents the respective material I or II. For the decisive case (interface condition is evaluated, i.e. p_{entry}^{II} is reached) $\lambda^* = \lambda^{II}$ holds, which means that we consider

$$\frac{\partial k_{rw}(S_{w,virt})}{S_w}(S_w = 1) = \frac{2 + 3\lambda^{II}}{\lambda^{II}} \left(\frac{p_d^{II}}{p_d^I} \right)^{2+3\lambda^{II}} \cdot \frac{\lambda^{II}}{\lambda^I}. \tag{50}$$

This derivative increases with

1. high values of the ratio $\frac{\lambda^{II}}{\lambda^I}$, especially with high λ^{II}, as λ^{II} also occurs in the exponent of the ratio of entry pressures, though a contrary effect is produced for the first term ($\frac{2+3\lambda}{\lambda}$ monotonously decreasing in $[0.2, \ldots, 3]$, the usual range of λ parameters)

2. a high ratio of entry pressures $\frac{p_d^{II}}{p_d^I}$.

According to the investigations made within the framework of this Master's Thesis, "trans numeric" always performs better than "trans exact" when realistic parameters for λ in $[0.2, \ldots, 3]$ are chosen.
To verify, that the numerical results of the newly constructed "trans exact" method are correct, "snapshots" of the saturation of the non-wetting phase at different simulation times are compared for the three methods. In Figures 32 through 34, the vertical non-wetting phase saturation profile is shown before the DNAPL front reaches the interface (920 s), when the DNAPL reaches the interface (1560 s), when the DNAPL is pooling up (1880 s), and a moment when the non-wetting phase has already entered (2990 s). To have an idea of the efficiency of these moments, the simulation times are marked by symbols in Figure 31.
Between the saturation fronts of the two methods with interface condition, no difference can be detected. As in these cases it is only the linearization which is different, the solutions should converge to the same value. The impression that for time step 3 in Figures 32 through 34 DNAPL has already entered the fine sand, is only due to the interpolation of the visualization tool. When comparing "standard" and "trans numeric" (or "trans exact") it becomes obvious that for the trans cases the DNAPL

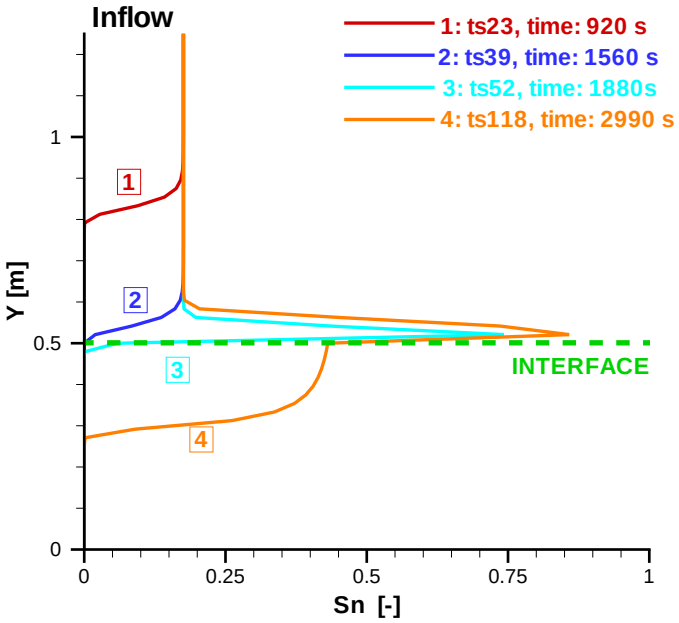

Figure 32: Vertical saturation profiles for "standard" at four chosen time steps

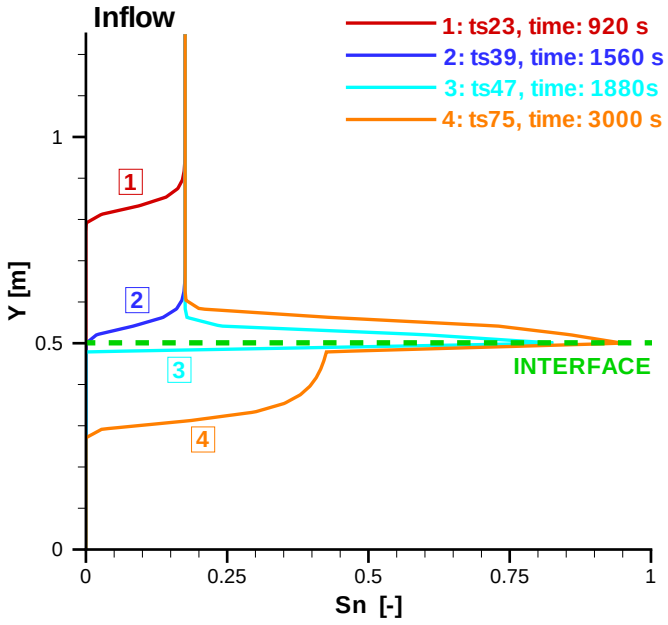

Figure 33: Vertical saturation profiles for "trans numeric" at four chosen time steps

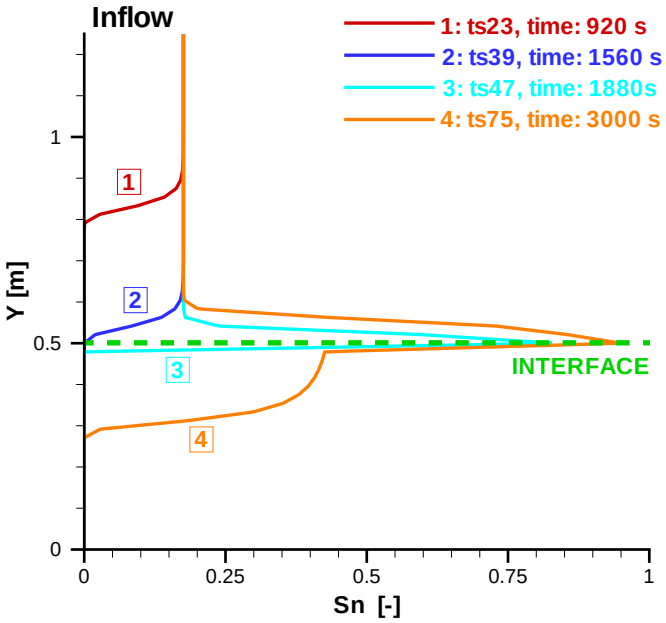

Figure 34: Vertical saturation profiles for "trans exact" at four chosen time steps

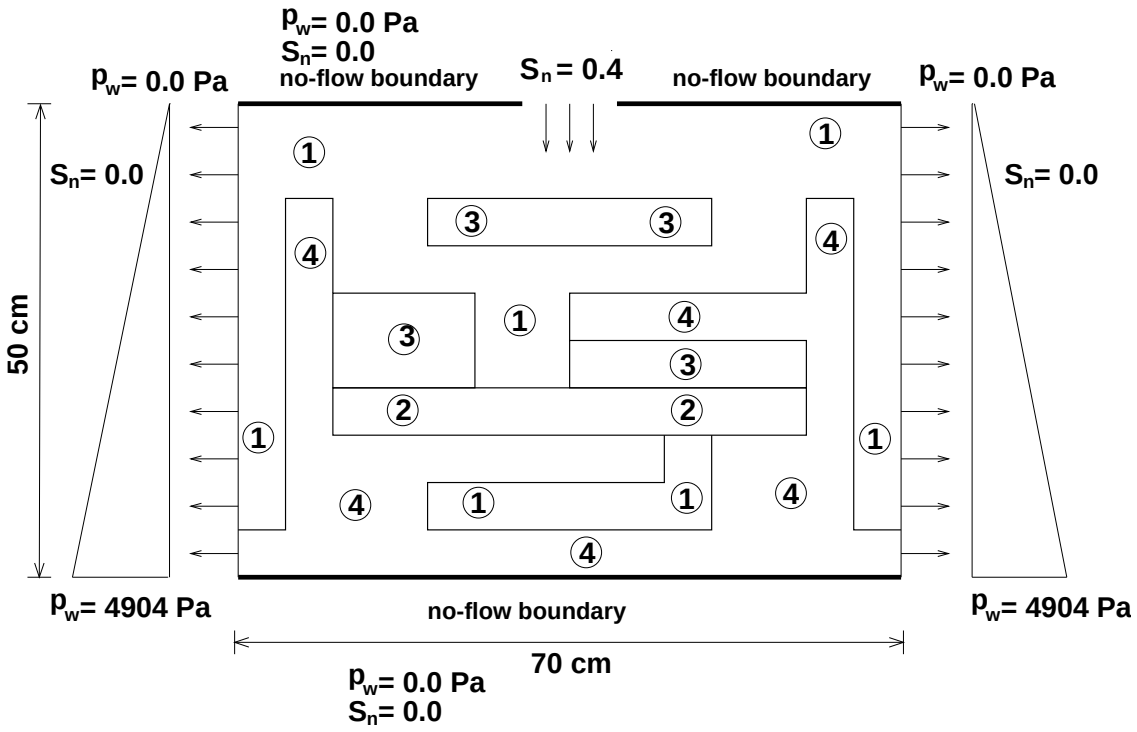

Figure 35: Setup of the experiment according to KÜPER

pools up up to higher values of the saturation at the interface. This is due to the fact that for the trans method the interface is correctly modelled and it is really necessary to reach the entry pressure before DNAPL can enter. For the standard scheme, the DNAPL enters too early (before the entry pressure is reached) so that the saturation at the interface remains lower.

5.2 A 2D example

In order to check the applicability of the three methods to an entirely realistic example, the experiment given in KÜPER [17] was modeled. KÜPER used an acrylic glass flume with the dimensions 70[cm] $\times$ 50[cm] $\times$ 0.6[cm]. In order to obtain a relatively complex layering, the flume is filled with four different kinds of homogeneous quartz sands which have uniform grain size distributions. Figure 35 shows the configuration of the sand layers.
The properties of the sands are given in Table 3, the fluid properties were the same as for the 1D experiments. The different permeabilities were determined by a hy-

draulic conductivity experiment and the capillary pressure - saturation relations for the DNAPL - water system were determined by a pressure cell experiment. For the parameterization, the BROOKS-COREY approach was applied.

	$p_d\,[Pa]$	$\lambda\,[-]$	$S_{wr}\,[-]$	$K\,[m^2]$	$\phi\,[-]$
sand 1	369.73	3.86	0.078	$5.041 \cdot 10^{-10}$	0.40
sand 2	434.45	3.51	0.069	$2.051 \cdot 10^{-10}$	0.39
sand 3	1323.95	2.49	0.098	$5.621 \cdot 10^{-11}$	0.39
sand 4	3246.15	3.30	0.189	$8.191 \cdot 10^{-12}$	0.41

Table 3: Sand properties for the KÜPER experiment

Except for the infiltration zone of the horizontal dimension of 10 cm, the upper and lower boundaries are impermeable. As in the previous experiment, the vertical boundaries were maintained at a constant hydrostatic pressure. The physical domain of the experiment shown in Figure 35 is discretized by a regular grid with a mesh size of $\Delta x = \Delta z = 0.0125$ m. The simulation started with an initial time step size of $\Delta t_{init} = 1$ s.
At the beginning of the DNAPL infiltration, the sand in the flume is fully saturated with water. The boundary conditions are shown in Figure 35. The DNAPL infiltration is controlled by a constant DNAPL saturation at the infiltration zone of $S_n = 0.4$.
The results of the computation (levels of DNAPL saturation) for three different simulation times are shown in Figures 36 to 38. As for the 1D VEGAS experiment, the performances at these time steps can be found by regarding the symbols in the respective illustration of the efficiency (Figure 39).
The impression that for the schemes with interface condition, DNAPL has entered sand 3 and 4 even at the beginning is only due to the interpolation done by the visualization tool. In fact, the only sand which can be entered by DNAPL is sand 2, and shortly before the end of the computation sand 3. The same differences as in the 1D example can be observed: for the two trans methods the DNAPL pools up to higher saturation levels when reaching sand 2 before finally entering. This is again due to the fact that for these methods the entry pressure has to be reached, before the non-wetting phase can enter the material with the higher entry pressure.
Investigating the efficiency for this 2D example shows, that also in this case the efficiency of "trans exact" for an example with realistic parameters does not reach the efficiency of the other methods - "trans exact" even performs very poorly The ratio of simulation time over computing time increases enormously starting from the moment when sand 3, which has a very high entry pressure, starts to be infiltrated. As the amount of DNAPL entered is still tiny, it cannot be detected visually, but only recog-

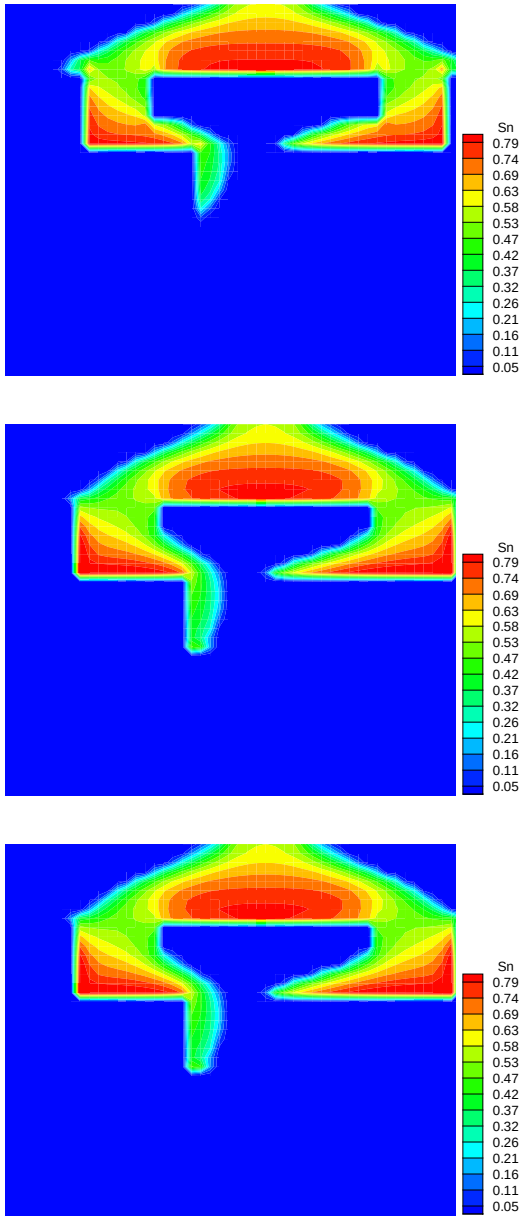

Figure 36: Saturation levels of TCE for the experiment after KÜPER et al. at $t = 184s$. Top: standard, middle: trans numeric, bottom: trans exact.

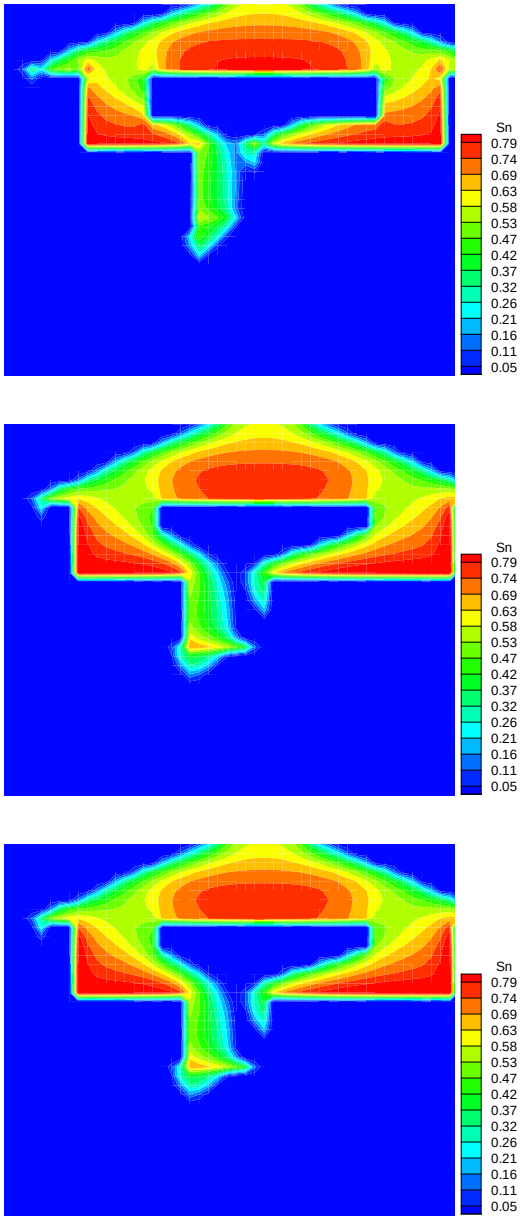

Figure 37: Saturation levels of TCE for the experiment after KÜPER et al. at $t = 220s$. Top: standard, middle: trans numeric, bottom: trans exact.

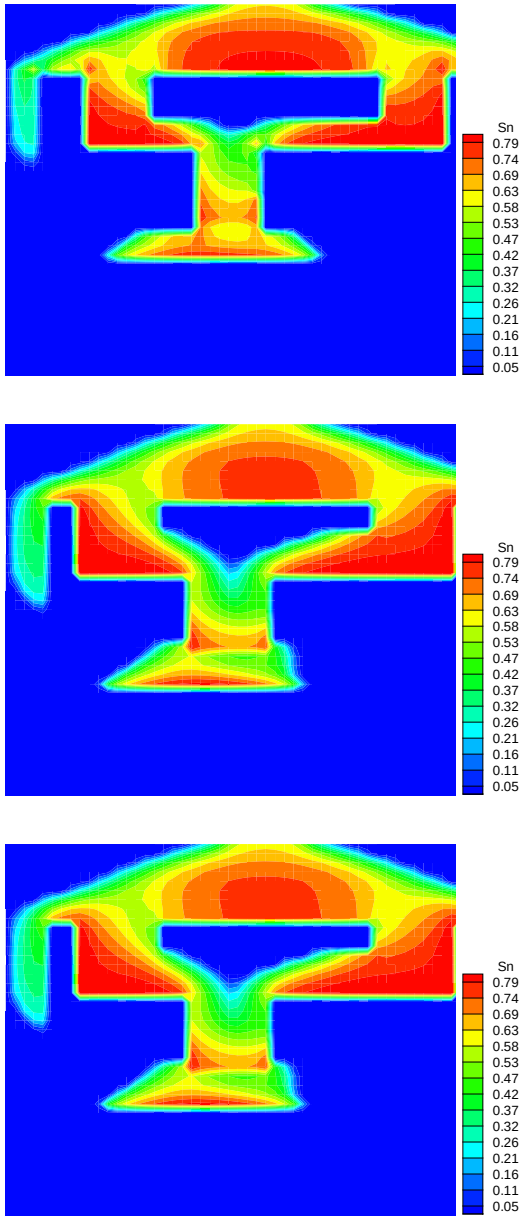

Figure 38: Saturation levels of TCE for the experiment after KÜPER et al. at $t = 312s$. Top: standard, middle: trans numeric, bottom: trans exact.

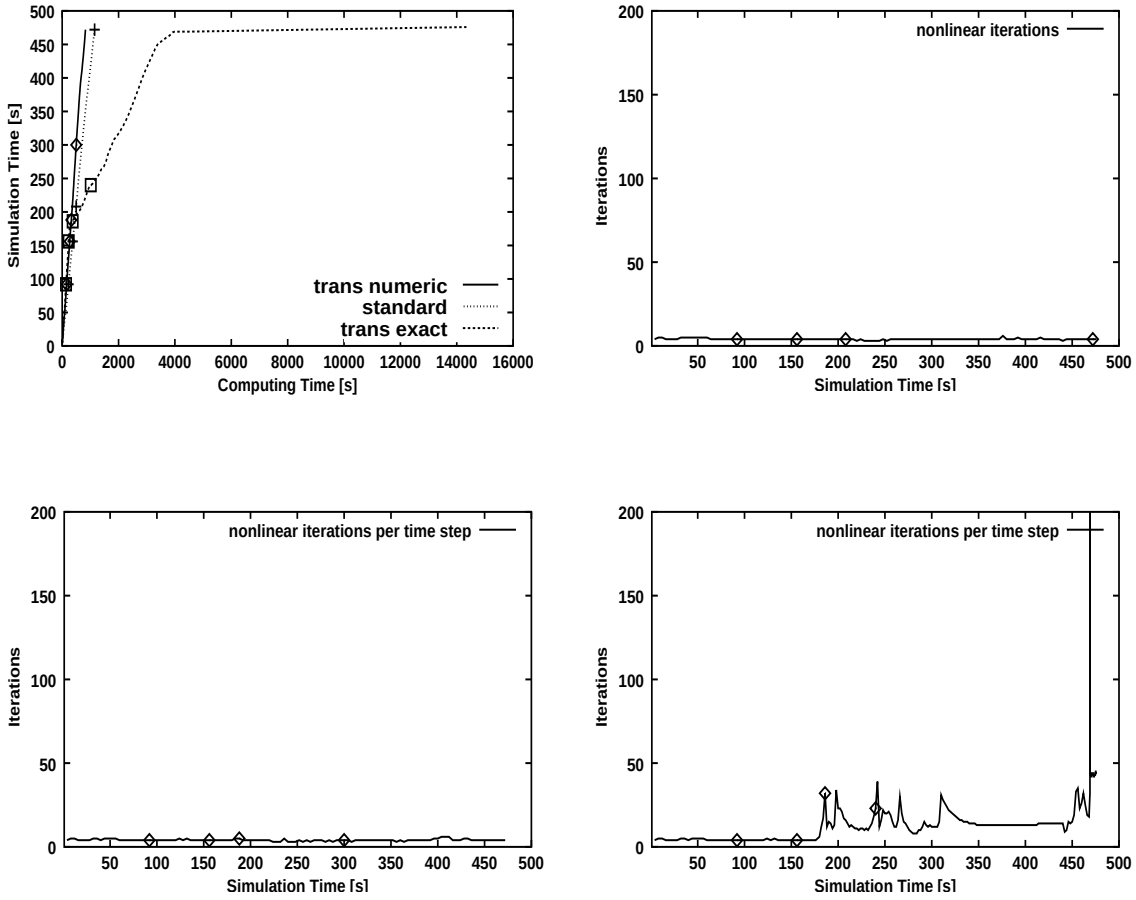

Figure 39: Simulation time over computing time and nonlinear iterations for the model setup according to the experiment of KÜPER. Upper right: "standard", bottom left "trans numeric", bottom right "trans exact".

nized from the numerical output of the simulation. In the maximum, more than 200 nonlinear iterations per time step can be detected, see Figure 39.

6 The Trust Region Method

6.1 Description and Applicability

In certain cases, the NEWTON method encounters difficulties. Two main indicators for problematic cases can be identified as

1. wrong search direction of the NEWTON method and
2. singularity of the global stiffness matrix.

NEWTON direction If the angle between the gradient and the NEWTON direction is large, which implies that the NEWTON algorithm searches for the solution in a wrong direction, the Trust Region algorithm can help to overcome these difficulties.
In order to find out if this is the case for the parameter sets, where the box scheme with interface condition and exact linearization performed badly, the global defect term $\mathbf{F}$ and the global stiffness matrix $\underline{\underline{G}}$ were computed for the simple example studied in Section 5.1. In the MUFTE code, the local stiffness matrix entries L^e_{ij} for each element are summed up to the global stiffness matrix $\underline{\underline{G}}$ according to the scheme shown in Figure 40 and in (51) for the simple 1D example of Subsections 5.1.1 and 5.1.2. The subscripts represent the local vertices and the superscripts the global elements. As we have to consider the derivatives of two equations ($f_w(p_w, S_n)$ and $f_n(p_w, S_n)$) for the two primary variables the matrix is split into four quadrants.

$$
\left[\begin{array}{ccccccc|ccc}
L^0_{00} & L^0_{01} & 0 & L^0_{03} & L^0_{02} & \dots & 0 & L^0_{04} & L^0_{05} & \dots \\
L^0_{10} & L^0_{11}+L^1_{00} & L^1_{01} & L^1_{13} & L^0_{12}+L^1_{03} & \dots & 0 & L^0_{14} & L^0_{15}+L^1_{14} & \\
0 & L^1_{10} & L^1_{11} & 0 & L^1_{13} & \dots & 0 & \vdots & & \\
L^0_{30} & L^0_{31} & 0 & L^0_{33}+L^2_{00} & L^0_{32}+L^2_{01} & \dots & 0 & & & \\
L^0_{20} & L^0_{21}+L^1_{30} & L^1_{31} & L^0_{23}+L^2_{10} & L^0_{22}+L^1_{33}+L^2_{11}+L^3_{00} & \dots & 0 & & & \\
\vdots & & & & & & & & & \\
\hline
L^0_{40} & L^0_{41} & \dots & & & & & L^0_{44} & L^0_{45} & \dots \\
L^0_{50} & L^0_{51}+L^1_{40} & & & & & & L^0_{54} & L^0_{55}+L^1_{41} & \\
\vdots & & & & & & & \vdots & &
\end{array}\right] \tag{51}
$$

For the investigations the, "critical" nonlinear iteration of the "critical" time step was chosen, i.e. the matrix where the interface condition was evaluated for the first time with p^{II}_{entry} reached, and the vector of the first and last line search corresponding to the critical nonlinear iteration.

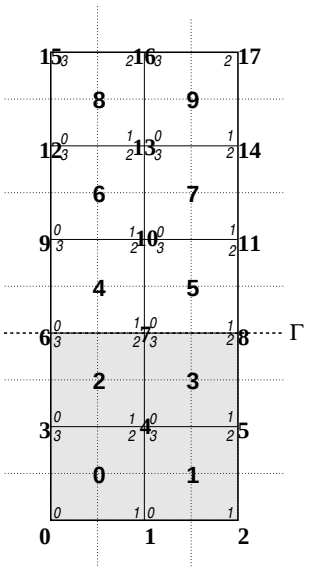

Figure 40: Denotations for local and global vertices

The angle Θ_k between the NEWTON direction $\mathbf{d_k}$ and the gradient $\mathbf{g_k}$ is computed according to

$$\Theta_k := \arccos \frac{< -\mathbf{g_k}, \mathbf{d_k} >}{\|\mathbf{g_k}\| \, \|\mathbf{d_k}\|} \tag{52}$$

where

- $\mathbf{d_k} = \underline{\underline{G}}_k^{-1} \mathbf{F_k}$
- $\mathbf{g_k} = \underline{\underline{G}}_k^{T} \mathbf{F_k}$

According to the investigations carried out, this angle Θ_k was indeed approximately 90° for the cases where "trans exact" encountered difficulties, which means that the NEWTON algorithm was searching in the wrong direction.

Singularity The global stiffness matrix is quasi-singular, if the ratio between the largest and the smallest singular value is high or especially if the smallest singular value is close to zero. For the studies carried out, common ranges of singular values lie in $[10^{-8}, 10^{6}]$ which in fact indicates quasi-singularity for the cases where the box scheme with interface condition and exact linearization performs badly.

6.2 Principles and Algorithm

Solving our nonlinear problem $\mathbf{F}(\mathbf{x}) = \mathbf{0}$ the NEWTON algorithm with trust regions converges to a stationary point of f, i.e. $\mathbf{grad}\, f(x_k) \to 0$ where $\{x_k\}$ is the generated sequence.

Principles: (taken from GILBERT [10], [9])

1. We are looking for stationary points of
$$f(x) = \frac{1}{2}\|\mathbf{F}(\mathbf{x})\|_2^2$$
2. We consider a quadratic model of f around x:
$$d \in \Re^n \mapsto \varphi(d) := \frac{1}{2}\|\mathbf{F}(\mathbf{x}) + \mathbf{F}'(x)d\|_2^2$$
3. φ is minimized ONLY in the trust region, i.e. the sphere $S(x, \Delta)$:
$$\begin{cases} min_d\ \varphi(d) \\ \|d\|_2 \leq \Delta \end{cases}$$
where
   - $\Delta > 0$ is called *trust radius*.
   - Δ has an influence on the displacement d.
   - Advantage: d is well defined even if $\mathbf{F}'(x)$ is singular.

Algorithm

Iteration from (x_k, Δ_k) to (x_{k+1}, Δ_{k+1}): We need constants independent of k...

- ... which measure the success: $0 < \eta_1 < \eta_2 < 1$
- ... to update Δ_k: $0 < \tau_1 \leq \tau_2 < 1 < \tau_3$.

1. Stop if already converged ($\|\mathbf{grad}\, f(x_k)\| \simeq 0$)
2. Calculation of the displacement d_k:
   (a) Calculate an approximate solution for d_k

(b) Calculate the concordance between the function f and its quadratic model φ:

$$\rho_k = \frac{f(x_k + d_k) - f(x_k)}{\varphi(d_k)}$$

(c) If $\rho_k < \eta_1$: reduce $\Delta_k \in [\tau_1 \|d_k\|, \tau_2 \|d_k\|]$ and go to 2a

3. Set $x_{k+1} = x_k + d_k$

4. Update trust radius:

$$\Delta_{k+1} \in \begin{cases} [\tau_2 \Delta_k, \Delta_k] & \text{if } \rho_k \leq \eta_2 \\ [\Delta_k, \tau_3 \Delta_k] & \text{else.} \end{cases}$$

This algorithm is used in the software N4IP1 called OPINeL [8], for example. It is based on the truncated NEWTON nonlinear interior point approach, developped by L. CHAUVIER and J. Ch. GILBERT.

7 Summary and Concluding Remarks

The investigations in this report have shown that the hoped for amelioration of efficiency of the box method with interface condition when the partial differential equations are linearised exactly is in general not obtained.
However, for certain cases for which the numerically linearized scheme with interface condition encountered particular problems (increase of nonlinear NEWTON iterations due to steep gradients of the relative permeability - saturation function of the wetting phase at $S_w = 1$) the efficiency can be improved by using exact linearization if favourable capillary pressure - saturation relationships are taken.
Taking the constitutive relationships of BROOKS-COREY and common ranges of the parameters, however, the trans scheme with exact linearization does not perfom as well as that with numerical linearization as in this case the numerical linearization has the effect of reducing the steepness in the gradients of the constitutive relationships.
To sum up, one can say that except for special applications the formerly used box scheme with interface condition and numerical linearization is more efficient than the scheme with exact linearization developped here. A possibility to improve the scheme in cases where the trans method with consistent linearization encounters problems, could be the use of the Trust Region algorithm.

A Analytical Derivatives of $\frac{\partial k_{r\alpha}(S_{n,virt})}{S_n}$ for Polynomial $k_{rw} - S_w$ and Brooks-Corey $p_c - S_w$ Relationships

The analytical differentiation of the relative permeability - saturation relationships is used in case we are at the interface, we consider an element which belongs to the material with the higher entry pressure and the entry pressure of this material has been reached.
As the derivatives for S_w are the same as for S_n except for the sign, the mobility derivatives are deduced here by differentiating for S_w and changing the sign accordingly.
The constitutive relationships used here read as follows:

$$\begin{aligned} k_{rw}(S_w) &= S_w^e && (53)\\ k_{rn}(S_w) &= (1-S_w)^e && (54)\\ p_c^I &= p_d^I \cdot S_w^{-\frac{1}{\lambda^I}} && (55)\\ p_c^{II} &= p_d^{II} \cdot S_w^{-\frac{1}{\lambda^{II}}}, && (56) \end{aligned}$$

where e and λ are real parameters.
We can compute the virtual saturation $S_{w,virt}$ as a function of the saturations S_w as follows:

$$\begin{aligned} p_c^I(S_w) &= p_c^{II}(S_{w,virt}) \\ p_d^I \cdot S_w^{-\frac{1}{\lambda^I}} &= p_d^{II} \cdot S_{w,virt}^{-\frac{1}{\lambda^{II}}} \\ S_{w,virt} &= \left(pd^{II}/pd^I\right)^{\lambda^{II}} \cdot S_w^{\frac{\lambda^{II}}{\lambda^I}} && (57) \end{aligned}$$

Using this relationship for $S_{w,virt}$, the derivatives of $\frac{\partial k_{r\alpha}(S_{w,virt})}{\partial S_w}$ are calculated:

$$\begin{aligned} \frac{\partial k_{rw}(InvPc^{II}(Pc^I(S_w)))}{\partial S_n} &= -\frac{\partial k_{rw}(S_{w,virt})}{\partial S_w} = -\frac{\partial \left[\left(\frac{p_d^{II}}{p_d^I}\right)^{\lambda^{II}} \cdot S_w^{\frac{\lambda^{II}}{\lambda^I}}\right]^e}{\partial S_w} \\ &= -e \cdot \left(\frac{p_d^{II}}{p_d^I}\right)^{\lambda^{II} e} \cdot \frac{\lambda^{II}}{\lambda^I} \cdot S_w^{\frac{\lambda^{II}}{\lambda^I}(e-1)} && (58) \end{aligned}$$

$$\begin{aligned}\frac{\partial k_{rn}(InvPc^{II}(Pc^{I}(S_w)))}{\partial S_n} &= -\frac{\partial k_{rn}(S_{w,virt})}{\partial S_w} = -\frac{\partial\left[1-\left(\frac{p_d^{II}}{p_d^{I}}\right)^{\lambda^{II}}\cdot S_w^{\frac{\lambda^{II}}{\lambda^{I}}}\right]^{e}}{\partial S_w} \\ &= -e\left[1-\left(\frac{p_d^{II}}{p_d^{I}}\right)^{\lambda^{II}} S_w^{\frac{\lambda^{II}}{\lambda^{I}}}\right]^{e-1}\frac{\lambda^{II}}{\lambda^{I}}\left(\frac{p_d^{II}}{p_d^{I}}\right)^{\lambda^{II}} S_w^{\frac{\lambda^{II}}{\lambda^{I}}-1} \quad (59)\end{aligned}$$

B Analytical Derivatives of $\frac{\partial k_{r\alpha}(S_{n,virt})}{S_n}$ for BROOKS-COREY $k_{rw} - S_w$ and $p_c - S_w$ Relationships

The analytical differentiation of the relative permeability - saturation relationships according to BROOKS and COREY is used in case we are at the interface, we regard an element which belongs to the material with the higher entry pressure and the entry pressure of this material has been reached.
As the derivatives for S_w are the same as for S_n except for the sign, the mobility derivatives are deduced here by differentiating for S_w and changing the sign accordingly.
The BROOKS-COREY relationships read as follows:

$$k_{rw}(S_w) = S_w^{\frac{2+3\lambda^*}{\lambda^*}} \tag{60}$$

$$k_{rn}(S_w) = (1-S_w)^2 \cdot \left(1 - S_w^{\frac{2+\lambda^*}{\lambda}}\right) \tag{61}$$

$$p_c^I = p_d^I \cdot S_w^{-\frac{1}{\lambda^I}} \tag{62}$$

$$p_c^{II} = p_d^{II} \cdot S_w^{-\frac{1}{\lambda^{II}}} \tag{63}$$

where λ^* stands for the λ parameter of the respective material, i.e. λ^I or λ^{II}.
We can compute the virtual saturation $S_{w,virt}$ as a function of the saturations S_w as follows:

$$\begin{aligned} p_c^I(S_w) &= p_c^{II}(S_{w,virt}) \\ p_d^I \cdot S_w^{-\frac{1}{\lambda^I}} &= p_d^{II} \cdot S_{w,virt}^{-\frac{1}{\lambda^{II}}} \\ S_{w,virt} &= \left(pd^{II}/pd^I\right)^{\lambda^{II}} \cdot S_w^{\frac{\lambda^{II}}{\lambda^I}} \end{aligned} \tag{64}$$

As for the polynomial relative permeability - saturation relationship (Appendix A), the derivatives of $\frac{\partial k_{r\alpha}(S_{n,virt})}{\partial S_n}$ are computed accordingly:

$$\begin{aligned} \frac{\partial k_{r\alpha}(InvPc^{II}(Pc^I(S_w)))}{\partial S_n} &= -\frac{\partial k_{rw}(S_{w,virt})}{\partial S_w} = -\frac{\partial \left[\left(\frac{p_d^{II}}{p_d^I}\right)^{\lambda^{II}} \cdot S_w^{\frac{\lambda^{II}}{\lambda^I}}\right]^{\frac{2+3\lambda^*}{\lambda^*}}}{\partial S_w} \\ &= -\frac{2+3\lambda^*}{\lambda^*}\left(\frac{p_d^{II}}{p_d^I}\right)^{\lambda^{II}\frac{2+3\lambda^*}{\lambda^*}} \cdot \frac{\lambda^{II}}{\lambda^I} \cdot S_w^{\frac{\lambda^{II}}{\lambda^I}\frac{2+3\lambda^*}{\lambda^*}-1} \end{aligned} \tag{65}$$

$$\begin{aligned}\frac{\partial k_{rn}(InvPc^{II}(Pc^{I}(S_w)))}{\partial S_n} &= -\frac{\partial k_{rw}(S_{w,virt})}{\partial S_w} = -\frac{\partial (1-S_{w,virt})^2 \cdot \left(1 - S_{w,virt}^{\frac{2+\lambda^*}{\lambda}}\right)}{\partial S_w} \\ &= \left[2(1-S_{w,virt}) \cdot \left(1 - S_{w,virt}^{\frac{2+\lambda^*}{\lambda^*}}\right) + (1-S_{w,virt})^2 \cdot \right. \\ &\quad \left. \cdot \frac{2+\lambda^*}{\lambda^*} \cdot S_{w,virt}^{\frac{2+\lambda^*}{\lambda^*}-1}\right] \cdot \frac{\lambda^{II}}{\lambda^{I}} \left(\frac{p_d^{II}}{p_d^{I}}\right)^{\lambda^{II}} \cdot S_w^{\frac{\lambda^{II}}{\lambda^{I}}-1} \end{aligned} \tag{66}$$

C Evaluation of the Terms for the Consistent Linearization

As derived in Section 4.3 the terms in the following equations have to be evaluated in order to implement the consistent linearization into the code [11].

$$\underbrace{\mathbf{K}_{T_{\alpha e}}^{k+1,r;k,0}(\mathbf{u})}_{I} \cdot \underbrace{\Delta \mathbf{u}_e^{k+1,r+1;k+1,r}}_{II} = \underbrace{\mathbf{K}_{E_{\alpha e}}^{k+1,r}(\mathbf{u}) \cdot \mathbf{u}_e^{k+1,r}}_{III} - \underbrace{\mathbf{r}_{\alpha e}^{k+1,r;k,0;k+1,0}}_{IV} \tag{67}$$

with

$$\mathbf{f}_{\alpha e}^{k+1,r;k,0} = \mathbf{K}_{E_{\alpha e}}^{k+1,r} \cdot \mathbf{u}_e^{k+1,r} - \mathbf{r}_{\alpha e}^{k+1,r;k,0;k+1,0}$$

and

$$\mathbf{K}_{T_{\alpha e}}^{k+1,r;k,0} = \frac{\partial \mathbf{f}_{\alpha e}^{k+1,r;k,0}}{\partial \mathbf{u}_e^{k+1,r}},$$

The terms I through VI are calculated under the assumptions $\frac{\partial \rho_\alpha}{\partial S_\alpha} = 0$ and $\frac{\partial \rho_\alpha}{\partial p_\alpha} = 0$ as mentioned in Section 4.3.

C.1 Term I: Tangent Element Matrix

Here the entries of the tangent element matrix are calculated in the following way:

$$\frac{\partial \mathbf{f}_{\alpha e}^{k+1,r;k,0}}{\partial \mathbf{u}_e^{k+1,r}} = \underbrace{\frac{\partial \mathbf{f}_{\alpha e}^{k+1,r;k,0}}{\partial p_{we}^{k+1,r}}}_{(1)} + \underbrace{\frac{\partial \mathbf{f}_{\alpha e}^{k+1,r;k,0}}{\partial S_{ne}^{k+1,r}}}_{(2)} = \mathbf{K}_{T_{\alpha e}}^{k+1,r;k,0}$$

The terms (1) and (2) can be formulated as follows. The explanation of the abbreviations for vectors and matrices can be found in this appendix Section C.5.

- term (1) for phase w:

$$\begin{aligned} \frac{\partial \mathbf{f}_{we}^{k+1,r;k,0}}{\partial p_{we}^{k+1,r}} &= \theta \mathbf{A}_e^{k+1,r} + \frac{1}{\Delta t} \mathbf{B}_e^{k+1,r} \\ &\quad + \frac{\partial \mathbf{A}_e^{k+1,r}}{\partial p_{we}^{k+1,r}} \left[(1-\theta) p_{we}^{k,0} + \theta p_{we}^{k+1,r} \right] + \frac{\partial \mathbf{B}_e^{k+1,r}}{\partial p_{we}^{k+1,r}} \frac{p_{we}^{k+1,r} - p_{we}^{k,0}}{\Delta t} \end{aligned}$$

$$
\begin{aligned}
& -\frac{\partial \mathbf{C}_e^{k+1,r}}{\partial p_{we}^{k+1,r}} \frac{S_{ne}^{k+1,r} - S_{ne}^{k,0}}{\Delta t} - \frac{\partial \bar{\mathbf{g}}_{we}^{k+1,r}}{\partial p_{we}^{k+1,r}} \\
= \;& \theta \mathbf{A}_e^{k+1,r}
\end{aligned}
$$

- term (2) for phase w:

$$
\begin{aligned}
\frac{\partial \mathbf{f}_{we}^{k+1,r;k,0}}{\partial S_{ne}^{k+1,r}} &= -\frac{1}{\Delta t}\mathbf{C}_e^{k+1,r} + \frac{\partial \mathbf{A}_e^{k+1,r}}{\partial S_{ne}^{k+1,r}} \left[(1-\theta)p_{we}^{k,0} + \theta p_{we}^{k+1,r}\right] - \frac{\partial \bar{\mathbf{g}}_{we}^{k+1,r}}{\partial S_{ne}^{k+1,r}} \\
&\quad + \frac{\partial \mathbf{B}_e^{k+1,r}}{\partial S_{ne}^{k+1,r}} \frac{p_{we}^{k+1,r} - p_{we}^{k,0}}{\Delta t} \\
&= -\frac{1}{\Delta t}\mathbf{C}_e^{k+1,r} + \frac{\partial \mathbf{A}_e^{k+1,r}}{\partial S_{ne}^{k+1,r}} \left[(1-\theta)p_{we}^{k,0} + \theta p_{we}^{k+1,r}\right] - \frac{\partial \bar{\mathbf{g}}_{we}^{k+1,r}}{\partial S_{ne}^{k+1,r}}
\end{aligned}
$$

- term (1) for phase n:

$$
\begin{aligned}
\frac{\partial \mathbf{f}_{ne}^{k+1,r;k,0}}{\partial p_{we}^{k+1,r}} &= \theta \mathbf{E}_e^{k+1,r} + \frac{1}{\Delta t}\mathbf{J}_e^{k+1,r} \\
&\quad + \left\{ \frac{\partial \mathbf{F}_e^{k+1,r}}{\partial p_{we}^{k+1,r}} \left[(1-\theta)S_{ne}^{k,0} + \theta S_{ne}^{k+1,r}\right] - \frac{\partial \mathbf{I}_e^{k+1,r}}{\partial p_{we}^{k+1,r}} \frac{S_{ne}^{k+1,r} - S_{ne}^{k,0}}{\Delta t} \right\} \\
&= \theta \mathbf{E}_e^{k+1,r}
\end{aligned}
$$

- term (2) for phase n:

$$
\begin{aligned}
\frac{\partial \mathbf{f}_{ne}^{k+1,r;k,0}}{\partial S_{ne}^{k+1,r}} &= \theta \mathbf{F}_e^{k+1,r} + \frac{1}{\Delta t}\mathbf{I_e}^{k+1,r} \\
&\quad + \frac{\partial \mathbf{F}_e^{k+1,r}}{\partial S_{ne}^{k+1,r}} \left[(1-\theta)S_{ne}^{k,0} + \theta S_{ne}^{k+1,r}\right] + \frac{\partial \mathbf{I}_e^{k+1,r}}{\partial S_{ne}^{k+1,r}} \frac{S_{ne}^{k+1,r} - S_{ne}^{k,0}}{\Delta t} \Bigg) \\
&\quad + \frac{\partial \mathbf{E}_e^{k+1,r}}{\partial S_{ne}^{k+1,r}} \left[(1-\theta)p_{we}^{k,0} + \theta p_{we}^{k+1,r}\right] \\
&\quad + \frac{\partial \mathbf{J}_e^{k+1,r}}{\partial S_{ne}^{k+1,r}} \frac{p_{we}^{k+1,r} - p_{we}^{k,0}}{\Delta t}
\end{aligned}
$$

$$
\begin{aligned}
&= \theta \mathbf{F}_e^{k+1,r} + \frac{1}{\Delta t}\mathbf{I_e}^{k+1,r} \\
&\quad + \frac{\partial \mathbf{F}_e^{k+1,r}}{\partial S_{ne}^{k+1,r}} \left[(1-\theta) S_{ne}^{k,0} + \theta S_{ne}^{k+1,r}\right] \\
&\quad + \frac{\partial \mathbf{E}_e^{k+1,r}}{\partial S_{ne}^{k+1,r}} \left[(1-\theta) p_{we}^{k,0} + \theta p_{we}^{k+1,r}\right]
\end{aligned}
$$

.

C.2 Term II: Vector of the Unknowns p_w and S_n

$$
\Delta \mathbf{u}_e^{k+1,r+1;k+1,r} = \mathbf{u}_e^{k+1,r+1} - \mathbf{u}_e^{k+1,r} = \begin{pmatrix} p_w \\ S_n \end{pmatrix}_e^{k+1,r+1} - \begin{pmatrix} p_w \\ S_n \end{pmatrix}_e^{k+1,r}
$$

with

$$
p_{we} = \begin{pmatrix} p_{w1} \\ p_{w2} \\ \vdots \\ p_{wn_{nodes}} \end{pmatrix}_e \qquad S_{ne} = \begin{pmatrix} S_{n1} \\ S_{n2} \\ \vdots \\ S_{nn_{nodes}} \end{pmatrix}_e .
$$

C.3 Term III: Vector $\mathbf{K}_{E_{\alpha e}}^{k+1,r} \cdot \mathbf{u}_e^{k+1,r}$ at the Last Known Iteration Step r at Time Step k+1

The term $\mathbf{K}_{E_{\alpha e}}^{k+1,r}$ contains all terms which are differentiated at the new time and iteration step $k+1$, and respectively r.

- phase w

$$
\begin{aligned}
(\mathbf{K}\cdot\mathbf{u})_w &= -\phi \frac{\mid B_i \mid}{\Delta t} S_n^{k+1,r} - \theta \int\limits_{\partial B_i} K \lambda_w d\Gamma_{B_i} p_w^{k+1,r} \\
\mathbf{K}_{E_{we}}^{k+1,r} \cdot \mathbf{u}_e^{k+1,r} &= -\left\{\theta \mathbf{A}_e^{k+1,r} + \frac{1}{\Delta t}\mathbf{B}_e^{k+1,r}\right\} p_{we}^{k+1,r} \\
&\quad + \left\{\frac{1}{\Delta t}\mathbf{C}_e^{k+1,r}\right\} S_{ne}^{k+1,r} \\
&= -\theta \mathbf{A}_e^{k+1,r} p_{we}^{k+1,r} \\
&\quad + \left\{\frac{1}{\Delta t}\mathbf{C}_e^{k+1,r}\right\} S_{ne}^{k+1,r}
\end{aligned}
$$

- phase n

$$
\begin{aligned}
(\mathbf{K}\cdot\mathbf{u})_n &= \phi\frac{\mid B_i \mid}{\Delta t}S_n^{k+1,r} - \theta\int\limits_{\partial B_i} K\lambda_n d\Gamma_{B_i} p_w^{k+1,r} - \theta\int\limits_{\partial B_i} K\lambda_n\frac{dp_c}{dS_n}S_n^{k+1,r}d\Gamma_{B_i} \\
\mathbf{K}_{E_{ne}}^{k+1,r}\cdot\mathbf{u}_e^{k+1,r} &= -\left\{\theta\mathbf{E}_e^{k+1,r}+\frac{1}{\Delta t}\mathbf{J}_e^{k+1,r}\right\}p_{we}^{k+1,r} \\
&\quad -\left\{\theta\mathbf{F}_e^{k+1,r}+\frac{1}{\Delta t}\mathbf{I}_e^{k+1,r}\right\}S_{ne}^{k+1,r} \\
&= -\theta\mathbf{E}_e^{k+1,r}p_{we}^{k+1,r} \\
&\quad -\left\{\theta\mathbf{F}_e^{k+1,r}+\frac{1}{\Delta t}\mathbf{I}_e^{k+1,r}\right\}S_{ne}^{k+1,r}
\end{aligned}
$$

.

C.4 Term IV

In $\mathbf{r}_{\alpha e}^{k+1,r;k,0;k+1,0}$ on the other hand the derivatives are evaluated at the old time and iteration step k, and respectively r. Moreover it contains the terms which are independent of the primary variables.

- phase w

$$
\begin{aligned}
\mathbf{r_w} &= \phi\frac{\mid B_i \mid}{\Delta t}S_n^{k,0} + (1-\theta)\int\limits_{\partial B_i} K\lambda_w d\Gamma_{B_i} p_w^{k,0} \\
&\quad + \int\limits_{\partial B_i} K\lambda_w\rho_w g d\Gamma_{B_i} - q_w \mid B_i \mid \\
\mathbf{r}_{we}^{k+1,r;k,0;k+1,0} &= -\left\{(1-\theta)\mathbf{A}_e^{k+1,r}-\frac{1}{\Delta t}\mathbf{B}_e^{k+1,r}\right\}p_{we}^{k,0} \\
&\quad -\left\{\frac{1}{\Delta t}\mathbf{C}_e^{k+1,r}\right\}S_{ne}^{k,0} \\
&\quad + \bar{\mathbf{g}}_{we}^{k+1,r} + \bar{\mathbf{m}}_{we}^{k+1,0}
\end{aligned}
$$

- phase n

$$
\mathbf{r_n} = -\theta\phi\frac{\mid B_i \mid}{\Delta t}S_n^{k,0} + (1-\theta)\int\limits_{\partial B_i} K\lambda_n d\Gamma_{B_i} p_w^{k,0}
$$

$$
\begin{aligned}
& +(1-\theta)\int\limits_{\partial B_i} K\lambda_n \frac{dp_c}{dS_n} d\Gamma_{B_i} S_n^{k,0} + \int\limits_{\partial B_i} K\lambda_n \rho_n g d\Gamma_{B_i} - q_n \mid B_i \mid \\
\mathbf{r}_{ne}^{k+1,r;k,0;k+1,0} &= -\left\{(1-\theta)\mathbf{E}_e^{k+1,r} - \frac{1}{\Delta t}\mathbf{J}_e^{k+1,r}\right\} p_{we}^{k,0} \\
& -\left\{(1-\theta)\mathbf{F}_e^{k+1,r} - \frac{1}{\Delta t}\mathbf{I}_e^{k+1,r}\right\}_e S_{ne}^{k,0} \\
& + \bar{\mathbf{g}}_{ne}^{k+1,r} + \bar{\mathbf{m}}_{ne}^{k+1,0},
\end{aligned}
$$

where $\bar{\mathbf{m}}_{ne}^{k+1,0}$ describes the mass flux over the boundary at at the unknown time step $(k+1)$.

C.5 Notation for some Matrices and Vectors

- phase w, nodes i and j, $j \neq i$, element e, time step $k+1$

$$
\begin{aligned}
\mathbf{A}_e &= \mathbf{A}_e(p_w) = \oint\limits_{\partial B_i} \left\{\lambda_{wije}^{k+1}\mathbf{K_e}\right\} \cdot \mathbf{n} d\Gamma_{B_i} \\
\mathbf{B}_e &= \mathbf{B}_e(p_w, S_n) = \left\{S_{nie}^{k+1} - S_{nie}^{k}\right\} \phi_e \frac{d\rho_{wije}^{k+1}}{dp_w} \mid B_i \mid = 0 \\
\mathbf{C}_e &= \mathbf{C}_e(p_w) = \phi \cdot \mid B_i \mid \\
\bar{\mathbf{g}}_{we} &= \bar{\mathbf{g}}_{we}(p_w) = \oint\limits_{\partial B_i} \left\{\lambda_{wije}^{k+1}\mathbf{K_e}\rho_{wie}^{k+1}\mathbf{g}\right\} \cdot \mathbf{n} d\Gamma_{B_i} \\
\bar{\mathbf{m}}_{we} &= \bar{\mathbf{m}}_{we}(S_n) = \oint\limits_{\partial B_i} \left\{\lambda_{wije}^{k+1}\mathbf{K_e}\,\mathbf{grad}\, p_w^{k+1} - \lambda_{wije}^{k+1}\rho_{wie}^{k+1}\mathbf{K_e}\mathbf{g}\right\} \cdot \mathbf{n} d\Gamma_{B_i} \\
& \quad + q_{we}^{k+1} \mid B_i \mid,
\end{aligned}
$$

- phase n, nodes i and j, $j \neq i$, element e, time step $k+1$

$$
\begin{aligned}
\mathbf{E}_e &= \mathbf{E}_e(S_n) = \oint\limits_{\partial B_i} \left\{\lambda_{nije}^{k+1}\mathbf{K_e}\right\} \cdot \mathbf{n} d\Gamma_{B_i} \\
\mathbf{F}_e &= \mathbf{F}_e(S_n) = \oint\limits_{\partial B_i} \left\{\lambda_{nije}^{k+1}\frac{dp_{ce}^{k+1}}{dS_{ne}^{k+1}}\mathbf{K_e}\right\} \cdot \mathbf{n} d\Gamma_{B_i} \\
\mathbf{I}_e &= \mathbf{I}_e(S_n) = \phi_e \cdot \mid B_i \mid + \left\{\phi_e S_{nie}^{k+1}\frac{d\rho_{ne}^{k+1}}{dp_{ne}}\frac{dp_{ce}}{dS_{ne}}\right\} \mid B_i \mid
\end{aligned}
$$

$$= \phi_e \cdot \mid B_i \mid$$

$$\begin{aligned} \mathbf{J}_e &= \mathbf{J}_e(S_n) = \phi_e S_{nie}^{k+1} \frac{d\rho_{ne}^{k+1}}{dp_{ne}} \mid B_i \mid \\ &= 0 \end{aligned}$$

$$\bar{\mathbf{g}}_{ne} = \bar{\mathbf{g}}_{ne}(p_w) = \oint_{\partial B_i} \left\{ \lambda_{nije}^{k+1} \rho_{nie}^{k+1} \mathbf{K_e g} \right\} \cdot \mathbf{n} d\Gamma_{B_i}$$

$$\begin{aligned} \bar{\mathbf{m}}_{ne} = \bar{\mathbf{m}}_{ne}(S_n) = \oint_{\partial B_i} \Bigg\{ & \lambda_{nije}^{k+1} \mathbf{K_e} \frac{dp_{ce}^{k+1}}{dS_{ne}} \mathbf{grad}\, S_{ne} + \lambda_{nije}^{k+1} \mathbf{K_e}\, \mathbf{grad}\, p_{we} \\ & - \lambda_{nije}^{k+1} \rho_{nie}^{k+1} \mathbf{K_e g} \Bigg\} \cdot \mathbf{n} d\Gamma_{B_i} + q_{ne}^{k+1} \cdot \mid B_i \mid \end{aligned}$$

D Variation of the virtual saturation $S_{n,virt}$ depending on the ratio of entry pressures $\frac{p_d^{II}}{p_d^I}$

As can be seen in Figure 41, the variation of the virtual variable δ' can be computed as

$$\delta' = S_{w,virt} - S_{w,inc,virt}. \tag{68}$$

The virtual saturation $S_{w,virt}$ is calculated in Appendix A as

$$S_{w,virt} = \left(\frac{p_d^{II}}{p_d^I}\right)^{\lambda^{II}} S_w^{\frac{\lambda^{II}}{\lambda^I}}. \tag{69}$$

With the help of the incremented non-virtual saturation

$$\begin{aligned} S_{n,inc} &= S_n + \epsilon(S_n + 1) \quad \text{with } \epsilon \text{ a small constant} \\ &= S_n(1+\epsilon) + \epsilon \\ &= (1 - S_w)(1+\epsilon) + \epsilon \\ S_{w,inc} &= 1 - S_{n,inc} \\ &= 1 - \epsilon - (1 - S_w)(1+\epsilon) \end{aligned} \tag{70}$$

we can calculate the incremented virtual saturation

$$S_{w,inc,virt} = \left(\frac{p_d^{II}}{p_d^I}\right)^{\lambda^{II}} \left[1 - \epsilon - (1 - S_w)(1+\epsilon)\right]^{\frac{\lambda^{II}}{\lambda^I}}. \tag{71}$$

Inserting equations (69) and (71) into equation (68) gives the variation of the virtual saturation as

$$\delta' = \left(\frac{p_d^{II}}{p_d^I}\right)^{\lambda^{II}} \left\{ S_w^{\frac{\lambda^{II}}{\lambda^I}} - \left[1 - \epsilon - (1 - S_w)(1+\epsilon)\right]^{\frac{\lambda^{II}}{\lambda^I}} \right\}, \tag{72}$$

which increases with increasing ratio of the entry pressures $\frac{p_d^{II}}{p_d^I}$.

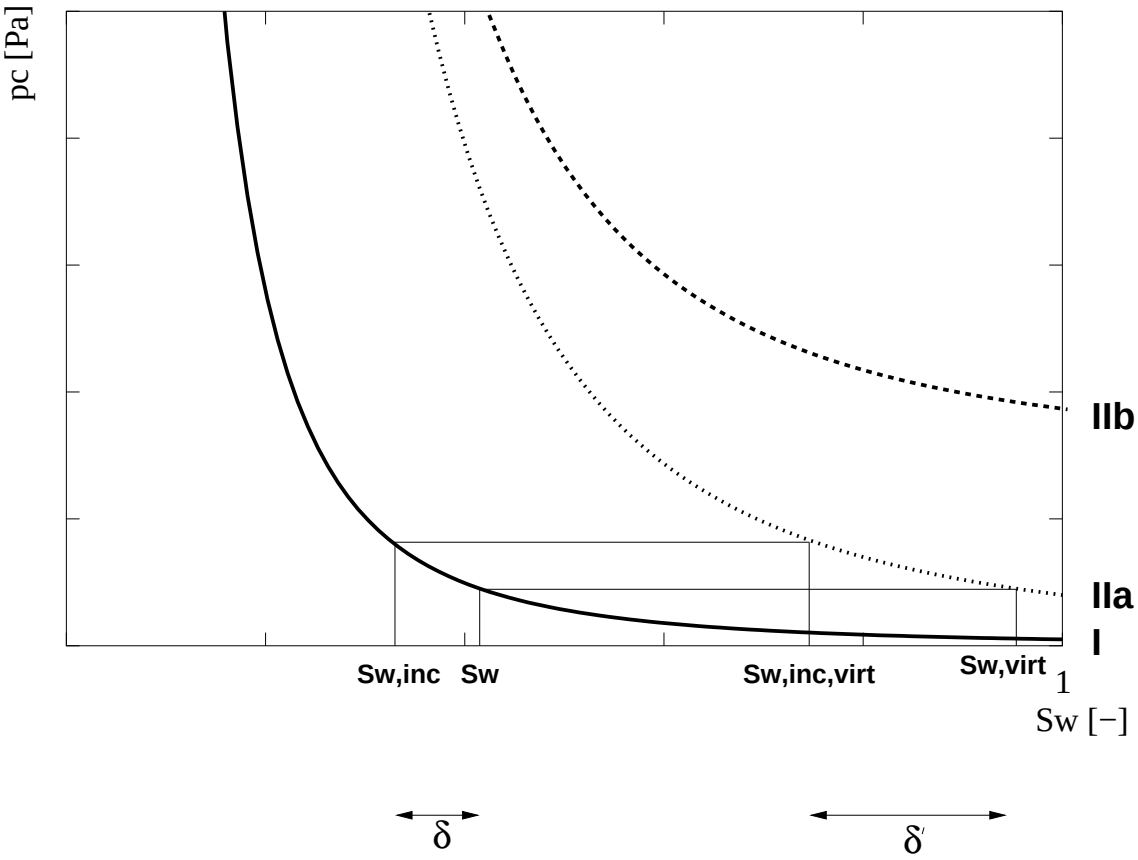

Figure 41: Definition of the variations δ and δ'

References

[1] Bastian, P. : Numerical computation of multiphase flows in porous media, habilitation thesis, Christian-Albrechts-Universität Kiel, 1999.

[2] Bielinski, A. : Numerische Modellierung von Strömungs- und Transportvorgängen für Mehrphasen-Mehrkomponenten-Systeme in geklüftet porösen Medien, Diplomarbeit, Institut für Computeranwendungen im Bauingenieurwesen, Technische Universität Braunschweig, 2001.

[3] Braun, J. : Ausbreitung von NAPL in gesättigten und ungesättigten Böden, in VEGAS workshop and BMBF/PWAB seminar – In-situ-Technologien zur Grundwasser- und Altlastensanierung, University of Stuttgart, 1996

[4] Burdine, N. : Relative permeability calculations from pore-size distribution data, Technical Report, Petroleum Transactions, AIME, 1953.

[5] Chavent, G. , and Jaffré, J. : Mathematical Models and Finite Elements for Reservoir Simulation, Studies in Mathematics and its Applications, Vol. 17, 1986.

[6] De Neef, M. J. : Modeling Capillary Effects in Heterogeneous Porous Media, PhD at the Technical University of Delft, 2000.

[7] Feldman, S.I. , Gay, D.M. , Maimone, M.W. and Schryer, N.L. : A Fortran-to-C Converter, Computing Science Technical Report No. 149, 1995.

[8] Fuduli, A. , and Gilbert, J.-Ch. : The module N4IP1 (nickname OPINel), user's guide, Version 0.2.11, 2003.

[9] Gilbert, J.-C. : Topics in Numerical Nonlinear Optimization, Inria-Rocquencourt, http://www-rocq.inria.fr/∼gilbert/ summer-school-jll/slides.html.

[10] Gilbert, J.-C. : Méthodes à régions de confiance, Inria-Rocquencourt, Extrait d'Optimisation Différentiable: Théorie et Algorithmes, Notes de Cours de l'ENSTA, version du 14 avril 2003.

[11] Helmig, R. : Theorie und Numerik der Mehrphasenströmungen in geklüftet-porösen Medien, Bericht Nr. 34, Institut für Strömungsmechanik und Elektron. Rechnen im Bauwesen der Universität Hannover, 1993.

[12] Helmig, R. : Multiphase flow and transport processes in the subsurface – a contribution to the modeling of hydrosystems, Springer Verlag, 1997.

[13] Helmig, R. and Huber, R. : Comparison of Galerkin-type discretization techniques for two-phase flow in heterogeneous porous media, Advances in Water Resources, 21: 697-711, 1998.

[14] Jaffré, J. , Roberts, J. E. , Wang, X. : Generalized cell-centered finite volume methods for two-phase flow in a porous medium with two rock types, Inria-Rocquencourt, BP 105, 78153 Le Chesnay Cédex, France, in preparation.

[15] Jakobs, H. : Auswirkungen der erweiterten Grenzflächenbedingung auf das numerische Verhalten eines Programms zur Simulation von Zweiphasen-Strömungen, Vortrag im Rahmen von SFB 404, Universität Stuttgart, 27.11.2002.

[16] Jakobs, H. , Helmig, R. , Miller, C. T. , Class, H. , Hilpert, M. , and Kees, C. E. : Multiphase flow and transport modeling in heterogeneous porous media, preprint submitted to Elsevier Science, 2003.

[17] Kueper, B.H. and Abbott, W. and Farquhar, G. : Experimental Observations of Multiphase Flow in Heterogeneous Porous Media, Journal of Contaminant Hydrology, 1989.

Unité de recherche INRIA Rocquencourt
Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)

Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38330 Montbonnot-St-Martin (France)
Unité de recherche INRIA Sophia Antipolis : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399